

# מקרו למתקדמים בלוטוס

## Intra-Macro -ו Meta-Macro

מוקדש באהבה להורי

לאה ולוי אברהמי

# מקרו למתקדמים בלוטוס

## Intra-Macro -ו Meta-Macro

רן אברהמי

ערך יצחק עמיהוד

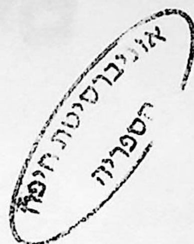
ספרי לימוד והכשרה במדעי המחשב
הוצאת הוד עמי

# Advanced Macro Writing in Lotus

**Meta-Macro and Intra-Macro**

**by Ran Avrahami**

edited by I. Amihud







**הוצאת הוד-עמי**
**לספרי מחשבים בע"מ**

ת.ד. 560, רמת-גן 52105
טל המשרד: 052-541207





הודפס בישראל
שבט תשנ"א, פברואר 1991


**Hod-Ami Ltd**
P.O.B. 560 Ramat-Gan
Israel, February 1991

ISBN 965-361-013-9 מסת"ב

# תוכן העניינים

## פרק 4: ספריית שירות מסוג mlb



## פרק 5: השליטה בדיסק



## פרק 6: ניהול קבצים ואבטחת מידע

# הקדמה

הספר הזה נולד במחשבתי כהמשך טבעי לסדנאות המומחים למשתמשי לוטוס שפיתחתי. ההוראה בסדנאות הוליכה אותי לפיתוח קורס הכולל את כל חומר הסדנאות והכנת הקורס הובילה אותי לכתיבת ספר זה. לפיכך רק טבעי הוא שדברי המבוא לקורס יהיו גם דברי המבוא לספר.

הספר, כמו סדנאות המומחים והקורס, נועד למשתמשי לוטוס ותיקים ואני מנסה לתת בו יותר מאשר "פטנטים" בהפעלת הלוטוס. ברצוני להקנות דרכי חשיבה שיעזרו למשתמש הלוטוס לעבור מדרגה, לגלות אפשרויות נוספות בלוטוס ובמיוחד כיצד אפשר בעזרתו לפרוץ אל מעבר לגבולות הגיליון וסרגליו וכמו "לשאוב אל הגיליון" את הדיסק כולו.

הקשר השוטף עם משתמשי לוטוס מתקדמים היה הדחף המרכזי לפיתוח הסדנאות הללו. הסתבר לי כי החלפת ידע פתוחה ומלאה בין משתמשים מניבה תועלת הדדית והנאה אישית לכל המשתתפים. מחשבה זו שיחררה אותי מהנטיה הטבעית "לשמור את כל הסודות לעצמי". ומשהגעתי לכתיבת הסדנאות והספר, נמצאתי לומד לא פחות מאשר מלמד.

יש מספר מרשים של משתמשי לוטוס ברמת תיחכום גבוהה ונראה לי שכולנו נצא נשכרים, אם תמצא דרך לתקשורת כמעט בין-אישית ברמת המשתמש המתקדם.

<u>קריאה מוקדמת:</u>

מומלץ שהקורא יכיר היטב את ה-manual של תוכנת לוטוס, גירסה 2.2. ויהיה לו ניסיון מוקדם בכתיבת מקרו. כך יוכל להפיק את מלוא התועלת מן הלימוד בספר זה.

כל הדוגמאות המוצגות בספר מבוססות על גירסה 2.2 של לוטוס. הן מתאימות גם לגירסה 2.01 (בשינויים מסוימים) ולגירסאות גבוהות יותר של לוטוס.

רן אברהמי

# חשיבה אינטגרטיבית בהפעלת גיליון לוטוס

הנושאים הנלמדים בקורס זה נבחרו בקפידה. הם מתייחסים לבעיות העומדות בפני משתמשים רבים תוך הצגת פתרונות, שרבים לא היו מודעים להם קודם.

הפעלת לוטוס נעשית בדרך כלל במישור ה"**תוך גיליוני**", כלומר באמצעות קובץ עצמאי, יחיד ונפרד. לרבים מתמקדת העבודה בגיליון בעולם ש"בין הסרגלים שעל המסך" תוך התנתקות מהעובדה שתוכנת לוטוס מאפשרת קשר דינמי עם מרחב המידע הענק שבדיסק. שימוש מתקדם בגיליון משמעותו שליטה מלאה של המשתמש בכל סביבת המיחשוב האישי שלו - **מרחב** המחשב.

כדי ליעל את השליטה והבקרה בייצור התפוקות של גיליון לוטוס דרושה **חשיבה אינטגרטיבית** בשני המישורים הללו - המישור ה**תוך גיליוני** והמישור ה**מרחבי**.

לחשיבה האינטגרטיבית במישור ה**תוך גיליוני** אנחנו קוראים **Intra-Macro** שכן היא מוליכה לשליטה אוטומטית בגיליון על ידי אינטגרציה של כל המרכיבים הקשורים בו. לחשיבה האינטגרטיבית במישור ה**מרחבי** אנחנו קוראים **Meta-Macro**, משום שהיא מוליכה מתוך הגיליון אל מעבר לו - לשליטה בכל מרחב המחשב.

אנו נשלב שני עקרונות אלו, **Meta-Macro** ו- **Intra-Macro**, בהקמת יישומי לוטוס. כך נקבל יישומים רבי עוצמה במבחר רב של נושאים ובמגוון רחב של תפוקות, תוך ניצול מירבי של כל הטמון בגיליון האלקטרוני.

**Meta-Macro ו- Intra-Macro הם פיתוח של רן אברהמי MBA LAC.**

בסוף הספר תמצא פירוט של נושאי הקורס ושאלון לקורא.

# פרק 1

# ניהול סביבת העבודה

## 1. חשיבה אינטגרטיבית

כשגיליון לוטוס "עולה" אל המסך מופיע לעינינו מבנה הסרגלים שלו, המפגש בין הסרגל העליון לבין הסרגל האנכי בפינה השמאלית העליונה של מטריצת ענק. התמונה הזאת ממקדת את העין ואת המחשבה אל כל הדברים הרבים שאפשר לבצע בגיליון. באותו הזמן אנחנו מודעים לכך שכל מה שנבצע במטריצה הזאת נשמור אחר כך בדיסק בכתובת כל שהיא המזוהה בשם בן שמונה תווים ובסיומת wk1. יצרנו "עולם סגור". עולם עשיר במיוחד במגוון האופציות שהוא מציע ליצירת יישומים ולשליטה במחשב. בכדי להפיק את מירב התועלת מאפשרויות אלו אנו נזקקים לדרך חשיבה מיוחדת המקנה ללוטוס משמעות של **שפה,** להבדיל **מאוסף של פקודות ופונקציות.**

שפה מורכבת מיחידות מודולריות ומחוקיות מסוימת של הקשרים בין היחידות הללו. כל שילוב של יחידות יש לו משמעות משלו, ואותן יחידות בשילוב שונה תקבלנה משמעות שונה. בלוטוס יש למעלה מ-500 פקודות תפריט (menu commands), עשרות פונקציות @ ועשרות פקודות מקרו. כל אחת ממאות היחידות הללו אפשר לשלב כמעט עם כל אחת מהיחידות האחרות ולקבל אין ספור צירופים אשר לכל אחד מהם משמעות שונה. אפשר לומר בצורה מטפורית, שכל יחידה ממאות היחידות הללו היא מלה, וכל צירוף מלים הוא משפט. וכאשר צירופי המלים למשפטים, פיסקאות וחיבורים שלמים שומרים על חוקיות מסוימת - יש לנו שפה. בדרך זו נקבל **מערך אינטגרטיבי של מרכיבים מודולריים.** מערך זה אני מכנה בשם Intra-Macro.

משפטים יוצרים תפוקות. תפוקתו של משפט היא המשמעות המועברת באמצעותו מהדובר אל חברו. גם למשפט Intra-Macro תפוקה משלו. תפוקה זו היא **תגובת המחשב** למשפט. לכל משפט מגיב המחשב בצורה שונה, אך שלא כבני אדם, תמיד יגיב המחשב לאותו משפט באותה דרך. כלומר, **לכל משפט Intra-Macro יש תפוקה אחת בלבד.**

תקשורת בין אנשים כוללת דיבור וכתיבה. תקשורת בין אנשים ומחשבים הינה באמצעות כתיבה בלבד. הכתיבה מאלצת את השפה למסגרות מסוימות - נייר, עיפרון, שורות, משבצות ואביזרים שונים נוספים. לכתיבה גם חוקים מיוחדים שנועדו לארגן את הנכתב, כך שתהיה לו מירב המשמעות. בלוטוס, שטח העבודה הוא הגיליון וכדי להקנות משמעות מירבית לשפה הנכתבת בו צריך לארגן אותו במבנה כל שהוא. מבנה זה הוא אחד

המרכיבים המודולריים של שפת ה-Intra-Macro, שכן זו שפה כתובה ולא שפה מדוברת.

שפת ה-Intra-Macro היא מכשיר לתקשורת בין האדם למחשב ומשתמשים בה לכתיבה בגיליון לוטוס. אותו "עולם סגור" בין הסרגלים, עולם שאני מתייחס אליו במונח "המימד התוך גיליוני" בחשיבה על התקשורת ביני לבין המחשב. אלא שבמהות המשפט "תקשורת ביני לבין המחשב" טמונה משמעות רבה - פריצה אל מעבר לעולם הסגור הזה, אל מעבר לגיליון.

מעבר לגיליון נמצא מרחב עבודה ענק המוגבל רק על ידי יכולתו הטכנולוגית של האדם. הדיסק הוא מרחב מסוג מיוחד המשמש לאחסנת נתונים. גודלו של מרחב האחסנה הזה לא מותנה בגודלו הפיסי-מכני של משטח הדיסק, אלא בכמות הנתונים שהטכנולוגיה של אותו הרגע יכולה לשמור בראשה של סיכה, וכמות זו הולכת וגדלה מיום ליום. בהשאלה אפשר לומר, כי כשם שהחלל שמסביב לכדור הארץ הוא אינסופי, כך גם החלל שבתוך המחשב ו"מסביב לגיליון" הוא אין סופי והגבול הנראה לעין הוא תוצר הטכנולוגיה של אותו הזמן. כך יכול גיליון הלוטוס להיות בסיס לשליטה בנפח הגדול הזה של הדיסק, אשר אתייחס אליו במונח "המימד המרחבי". את דרך החשיבה המקימה את הגשר בין המימד התוך גיליוני לבין המימד המרחבי אכנה בשם **Meta-Macro**.

מיפוי היא דרישה מקדימה לשליטה במרחב ומשיש מפה אפשר לקבוע כתובות. לגבי, שם הקובץ הוא כתובתו במפת Meta-Macro. קבצים שכתובתם מקבלת את הסיומת wk1 נמצאים ב"שכונה" מיוחדת במרחב התפישתי (קונספטואלי) של הדיסק, "שכונת הלוטוס". המייחד את הקבצים הללו מאחרים הוא המבנה שלהם. יש איזורים נרחבים מסביב ה"מאוכלסים" בקבצים בעלי מבנה שונה, כמו למשל מבנה ASCII. רוב התוכנות המקובלות בשוק מסוגלות לייבא ולייצא קובץ ASCII "נקי".

פן אחד של הלוטוס שנותר חבוי מעיני רבים הוא יכולתו הרבה במחרוזות טקסט ובשליטה בקובצי ASCII. יכולת זו נובעת מפונקציות String שלו, מן היכולת לשלוט בקובצי ASCII שלמים באמצעות פקודות התפריט ומשליטתו ברשומות **ובתווים בודדים** בדיסק באמצעות פקודות המקרו השונות. פן נוסף הוא יכולתו לאפשר אוטומציה מלאה של פעילות מחזורית, אשר ניתן להגדירה בדקות ספורות ואשר חוסכת אחר כך זמן וטעויות. התוצאה הינה יכולת שליטה כמעט בלתי מוגבלת **באמצעות הגיליון** בקובצי ASCII שבדיסק. תוצר הלוואי של שליטה זו היא היכולת לניהול תקשורת בין תוכנות "שכנות" על ידי העברת נתונים מאחת לחברתה תוך כדי ביצוע עיבודים, כאשר הנתונים הללו "עוברים" דרך גיליון לוטוס.

כשמשלבים את שני העקרונות הללו, עיקרון ה-**Meta-Macro** ועיקרון ה-**Intra-Macro**, בהקמת יישומי לוטוס, מקבלים יישומים שיהיו להם התכונות הבאות:

1. היישום יהיה פשוט ו"חכם".

2. היישום יבצע מספר רב של מטלות במספר קטן ביותר של פקודות מקרו.

3. היישום יאפשר שליטה במרחב המחשב שמעבר ללוטוס - דרך הלוטוס.

4. המבנה המודולרי מאפשר:

1. שימוש חוזר במרכיבים רבים בתוך הגיליון ובין גיליונות שונים.
2. מספר רב ביותר של צירופים בין מרכיבי היישום.
3. יעילות רבה בהכנסת שינויים ביישום.
4. איתור מהיר וקל של טעויות.
5. שילוב פונקציות בתוך מקרו.

ספר זה מבוסס על סדנאות המומחים למשתמשי לוטוס שפיתחתי. הנושאים נבחרו בקפידה והם מקיפים מכלול בעיות העומדות היום בפני משתמשים רבים. חשיבה אינטגרטיבית לטיפול בבעיות אלו עשויה לפתוח אפשרויות רבות נוספות לפתרונות שלא היו מודעים להם קודם.

## 2. מיפוי הזיכרון

כל הקמה של מבנה מורכב דורשת תכנון והכנת תשתית. כך בהקמת בית וכך בהקמת יישום לוטוס. לכן, לפני שניגשים לייצר תפוקות בלוטוס רצוי לבצע מיפוי מקדים של הדיסק. אפשר לגשת למיפוי כזה מנקודות מבט שונות. הדרך המומלצת כאן מכינה תשתית נוחה לפעילות אינטגרטיבית של Meta-Macro.

מיפוי זה מתחיל בהכנת רשימת יישומים שלכל אחד מהם תכונות מובהקות המבדילות אותו מאחרים. למשל,

1. קובצי DOS.
2. קובצי הפעלה של לוטוס (בלי קובצי wk1).
3. קובצי bat להפעלה כ-Menu במסך DOS.
4. קובצי wk1 המספקים שירות לכל יישומי לוטוס.
5. מערך ראשי ליישומי לוטוס.
   1. סביבה ליישום לוטוס להפעלת סדנא 1.
   2. סביבה ליישום לוטוס להפעלת סדנא 2.
   :
   n סביבה ליישום לוטוס להפעלת סדנא n.
6. קובצי הפעלה של תוכנת שירות, למשל Magellan.
7. קובצי הפעלה של מעבד התמלילים.
8. קובצי תמליל שנכתב במעבד התמלילים.
9. קובצי הפעלה של תקשורת טלפונית.
10. קבצים שנתקבלו בתקשורת טלפונית.

לכל אחד מאלה נקצה "זהות" על מפת הזיכרון (Memory-Map). ה"מפה" הזאת תהיה התשתית הארגונית של הדיסק. יש לזכור שתשתית זו ניתנת לשינוי על ידי הוספות וביטול של "זהויות" על המפה לפי הצרכים המשתנים של כל משתמש ומשתמש.

לפני שאדגים מפה כזאת, נייחד כמה מלים על **menu** (תפריט) במסך DOS.

לכל יישום יש נוהל הפעלה משלו. למשל,

להפעלת הלוטוס [100;100]B ... cd\123
123

ולהפעלת מעבד התמלילים .... cd\wm
wm

קל מאוד להכין מספר קובצי bat שכל אחד מהם מכיל נוהל הפעלה של יישום אחד ואשר שמו n.bat כאשר n הוא מספר כל שהוא. את כל הקבצים הללו נרכז במחיצה אחת הנקראת Menu. קובץ אחד מתוכם שנקרא בשם Menu.Bat יקרין על המסך תוכן עניינים המראה את רשימת היישומים הראשיים שיש להם זהות מוגדרת במפת הזיכרון ואת מספריהם. מכאן רק צעד אחד מוליך אל אוטומציה נוחה של הכניסה למחשב, שלא דרך תוכנות עזר, אלא "תוצרת עצמית".

מוסיפים כך את השורות הבאות בקובץ Autoexec.Bat:

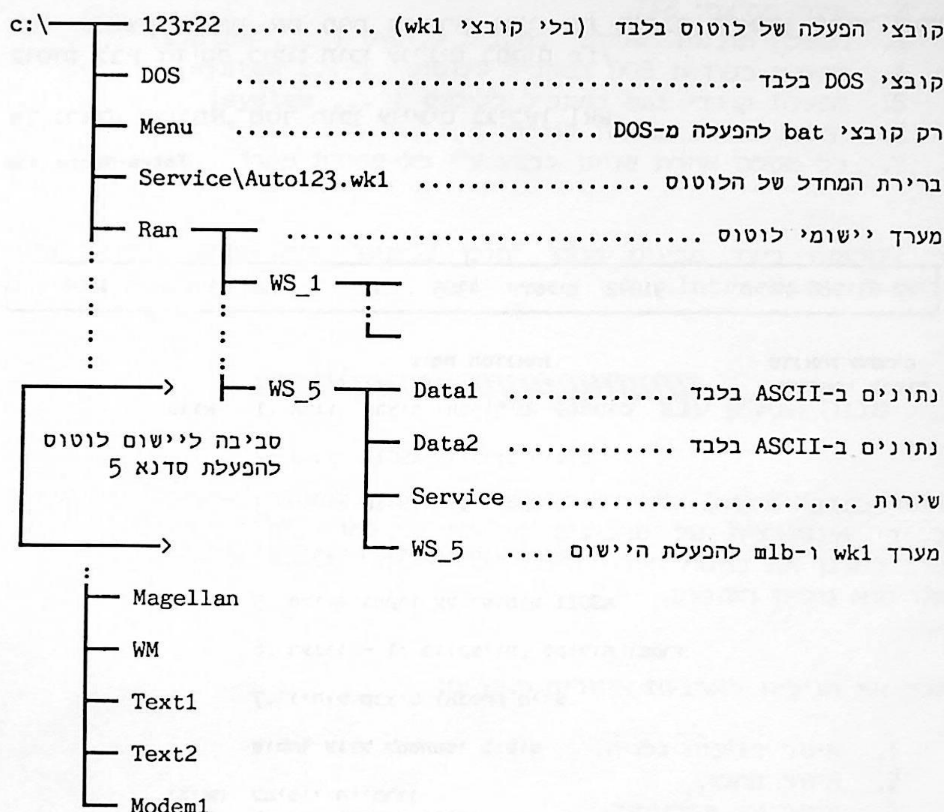
במקום כל שהוא בתחילת הקובץ ... Path = c:\;c:\DOS;c:\Menu;
prompt = p$g$
השורה שלפני האחרונה תהיה ...... cd\
והשורה האחרונה בקובץ זה תהיה ... Menu

שתי השורות האחרונות האלו תהיינה שתי השורות האחרונות בכל אחד מקובצי bat של תוכן העניינים.

התפוקה של הנוהל הזה:

1. לאחר פתיחת המחשב מופיע מסך תוכן העניינים הפרטי עם: c:\>
2. לאחר הקשת מספר מתוכן העניינים מופעל היישום השייך לו.
3. יציאה מכל יישום מחזירה את מסך תוכן העניינים ואת סימן ה-prompt: c:\>

עכשיו, הבה נראה מבט כללי של מפת זיכרון לתשתית כזאת.


Path = c:\;c:\DOS;c:\Menu;
c:\
123r22 .......... (wk1 קובצי הפעלה של לוטוס בלבד (בלי קובצי
DOS ......................... קובצי DOS בלבד
Menu ......................... רק קובצי bat להפעלה מ-DOS
Service\Auto123.wk1 ............ ברירת המחדל של הלוטוס
Ran ......................... מערך יישומי לוטוס
WS_1
WS_5
Data1 ............ נתונים ב-ASCII בלבד
Data2 ............ נתונים ב-ASCII בלבד
Service ........................ שירות
WS_5 .... מערך wk1 ו-mlb להפעלת היישום
סביבה ליישום לוטוס
להפעלת סדנא 5
Magellan
WM
Text1
Text2
Modem1

## 3. תוכן עניינים בלוטוס במקום fr/

לפני שנמשיך לבחון את מפת הזיכרון נבנה את הנדבך הראשון בקשר שבין לוטוס לבין הדיסק כולו: תוכן עניינים במקום fr/

כך נראה, לדוגמא, מסך תוכן עניינים בגיליון wk1.

**מפת Intra-Macro**

| (c) 1990 רן אברהמי LAC | ת.ד. 4305 ירושלים 91042 | טלפון 02-631768 |
|---|---|---|

| | צומת הסדנאות | סדנאות מומחים למשתמשי לוטוס |
|---|---|---|
| | ///////////////////////////////////////// | |
| סדנא | 1. ארגון קבצים ותפריטים | |
| | 2. קובצי ASCII וגיליון לוטוס | |
| | 3. הפעלת הגיליון ע"י תפריט | |
| | 4. שירות מתוך Macro Library | |
| | 5. שליטה במקרו על רשומות ASCII | |
| | 6. רענון - 1: פונקציות, פקודות ומקרו | |
| | 7. ניהול קבצים ואבטחת מידע | |
| | שולחן עגול למשתמשי לוטוס | |
| יציאה | למיפוי הזיכרון<br>למערכת ההפעלה (DOS)<br>לגיליון לוטוס חדש<br>מלוטוס | |

הסבר:

- הקווים האלכסוניים מסמנים את מיקום הסמן.
- הסרגלים מוחבאים על ידי מלת המקרו {frameoff}.
- בשורת הדו-שיח הראשונה מופיע לוגו (עד 80 תווים) על ידי {indicate Msg1}
- הקובץ במצב של CMD שהופעל ע"י מקרו \0
- כדי לבצע פעולה, או לקרוא לקובץ לפי שורה כל שהיא בתפריט, מביאים את הסמן על השורה המבוקשת ומקישים ENTER.
- אין הגבלה על מספר הפריטים להפעלה בתוכן עניינים כזה.

המטלות המתבצעות על ידי קובץ זה:

1. שליפת קובצי wk1 (במקום /fr)
2. שינוי מחיצה: /fd
3. הפעלת תוכנות "שכנות" במקביל ללוטוס { ... system}
4. הפעלת פקודות DOS במקביל ללוטוס { ... system}
5. הפעלת קובצי bat במקביל ללוטוס { ... system}
6. יציאה ל-DOS מגיליון לוטוס: /s
7. כל מטלה אחרת שניתן לבצע ע"י כל פקודת מקרו

כדי להראות כיצד מכינים קובץ "תוכן עניינים" כזה, נסקור תחילה את עיקרון ה-Intra-Macro.

## 4. מבנה הגיליון עבור Intra-Macro

המושג Intra-Macro מתייחס לחשיבה המוליכה לשליטה אוטומטית בגיליון על ידי אינטגרציה של מרכיבים מודולריים. כמו במשחק "לגו", גם כאן חשוב לארגן את משטח העבודה לפני שמתחילים להשתמש בו. משטח העבודה שלנו הוא גיליון הלוטוס.

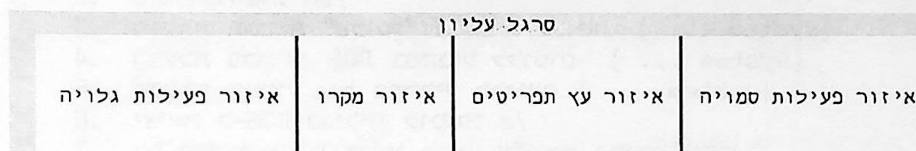
נחלק את הגיליון לארבעה איזורים ראשיים:

1. איזור פעילות גלויה.
2. איזור מקרו.
3. איזור עץ תפריטים.
4. איזור פעילות סמויה.

יש לשמור על מספר כללים בחלוקה זו של הגיליון:

1. כל איזור מתחיל בשורה העליונה הראשונה של הגיליון ונמשך עד לתחתיתו.
2. כל האיזורים מקבילים זה לזה. לא תהיה עמודה משותפת לשני איזורים.
3. אין מגבלה על מספר העמודות בכל איזור.

במהלך הפרקים הבאים יפורטו ויודגמו הפעולות המתבצעות בכל איזור ואיזור. בחלוקה סכימתית יראה הדבר כך:




## 5. Intra-Macro לשליפת קבצים

### 5.1 מבנה בסיסי

```
   A       B         C        D        E              F
1      תוכן עניינים                   \0      {?}/c{bs}{r 2}~xData1~
2      ///////                                 /fr
3                                   xData1         <---
4          1       סדנא      WS_1              ~       |
5                                                      |
6          2              [WS_2] ----------------------
```

הסבר:

1. כאשר הקובץ נטען לזיכרון הסמן מוצב ב-B2 והמקרו **\0** מכניס אותו מיד תחת CMD.

2. מלת המקרו {?} מאפשרת להזיז את הסמן בחופשיות תחת המקרו. נציב, למשל, את הסמן בתא **B6** (סדנא 2) ונקיש ENTER.

3. שם הקובץ WS_2 יועתק אל **xData1** - תחום מוגדר כמקום ריק בעמודת המקרו (שם המקרו כתוב בעמודה E והמקרו עצמו ב-F).

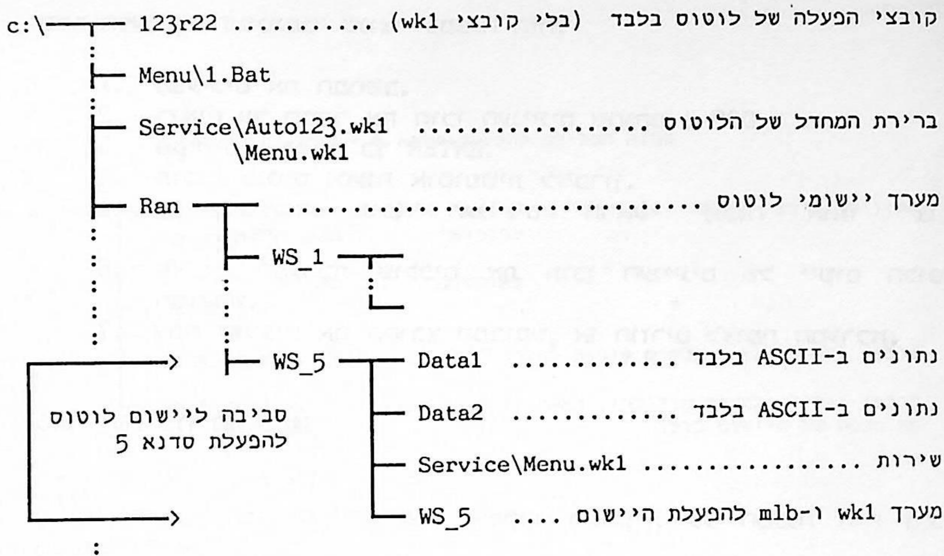
4. תתבצע פעולת השליפה.

### 5.2 מיקום תוכן העניינים במפת הזיכרון

קובץ תוכן העניינים ייקרא תמיד בשם Menu.wk1 והוא יישמר בתת-מדריך (Sub-Directory) בשם Service, פרט למספר חריגים. החשוב שבחריגים הללו קשור בכניסה האוטומטית למערך הלוטוס.

נתבונן שוב בחלק של מיפוי הזיכרון המתייחס למערך יישומי לוטוס.

## מיפוי הדיסק




במחיצה c:\123r22 מאוחסנות כל תוכנות ההפעלה של לוטוס גירסה 2.2, ואך ורק קבצים אלה. לא נאחסן כאן קובצי נתונים, קובצי wk1, או כל קובץ שאיננו שייך ישירות ללוטוס.

בעת התקנת תוכנת לוטוס בפעם הראשונה נבצע ~/wgddc:\Service\ ונעדכן. בכך נקבעה המחיצה c:\Service כברירת המחדל של תוכנת לוטוס הנמצאת במחיצה c:\123r22

במחיצה זו (c:\Service) נשמור את קובץ תוכן העניינים הראשי של כל מערכת יישומי הלוטוס תחת שני שמות: קובץ Menu.wk1 וקובץ Auto123.wk1 (הסיבה תתברר בהמשך).

במחיצה c:\Menu המכילה, כזכור, קובצי n.Bat [n = כל מספר] נכין קובץ בשם 1.Bat ובו נכתוב:

```
echo off
cls
cd\123r22
123
cd\
cls
Menu
```

בכל סביבה של יישום לוטוס נכללת מחיצה בשם Service. למשל המחיצה c:\Ran\Ws_5\Service מיועדת לסביבה של סדנא 5, וכך בכל סביבת יישום לוטוס אחרת.

סדר הפעולות האוטומטי שנוצר במבנה הזה:

1. מפעילים את המחשב.
2. רואים על המסך את תוכן העניינים שהוקם ב-DOS.
3. מקישים 1 ואחר כך ENTER.
4. תוכנת לוטוס נטענת אוטומטית לזיכרון.
5. מופיע מסך תוכן העניינים הראשי (נכנה אותו "צומת המערכת").
6. מצומת המערכת שולפים את תוכן העניינים של יישום הלוטוס המבוקש.
7. עתה שולפים את הקובץ המבוקש, או חוזרים לצומת המערכת.

## 5.3 שליפה מורכבת

ראינו למעלה מבנה בסיסי של **"המקרו השולף"**. על המבנה הזה נרכיב עכשיו תוספות.

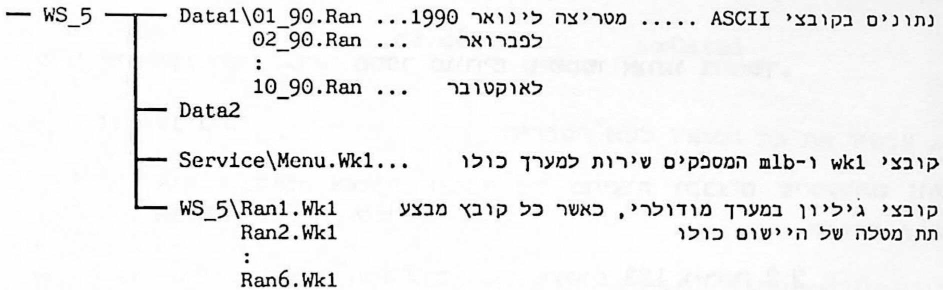
רצוי להיכנס לסביבת יישום הלוטוס המבוקש על ידי שינוי המחיצה של ברירת המחדל ואחר כך שליפת תוכן העניינים של אותה סביבה. למשל, כדי לעבור מצומת המערכת אל סדנא 5 נבצע ~fdc:\Ran\Ws_5\Service\/ ורק אחר כך נשלוף את תוכן העניינים של סדנא זו. לביצוע מטלה זו ייראה "המקרו השולף" כך:

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | | תוכן עניינים | | | \0 | {?}/c{bs}{r 2}xData1~ |
| 2 | | /////// | | | | /fr |
| 3 | | | | | xData1 | < |
| 4 | | 1 | סדנא | WS_1 | | ~ |
| 5 | | | | | | |
| 6 | | 5 | | {esc 6}/fdc:\Ran\WS_5\Service\~/frMenu | | |
| 7 | | | | | | |
| 8 | | | | | | |

כך גם ייראה המקרו שיחזיר אותנו מסדנא 5 אל צומת המערכת.

## 5.4 מיפוי תוך סביבתי

תוכן העניינים של סביבת היישום הספציפי מקשר בינינו לבין מפת הזיכרון באותו איזור. נבחן מקרוב את המיפוי התוך סביבתי של סדנא 5 לדוגמא.




את המיפוי המקדים ליישומי לוטוס יש להכין עד רמת תת-מדריך (Sub-Directory) בלבד. את הקבצים עצמם יוצרים תוך כדי עבודה בלוטוס. במפת סדנא 5 רואים שלושה סוגי מחיצות:

1. \WS_5\Service

מחיצה זו נועדה לאחסון Menu.wk1 ספציפי לסדנא 5 והיא תהיה ברירת המחדל כל זמן שסביבה זו תהיה פעילה. עם החזרה לצומת המערכת תחזור מחיצת ברירת המחדל ותהיה c:\Service כפי שתואר קודם.

\WS('(0_5\WS_52

מחיצה זו מאחסנת את כל קובצי wk1 המבצעים מטלות שונות במהלך הסדנא. כל אחד מהקבצים הללו מבצע סדרת פעולות שיש קשר מהותי ביניהן, כמו קטעים שונים בפרק אחד בספר. הכותרת המאפיינת את ה"פרק" הזה תופיע ב-Menu.wk1 של הסדנא ומימינה כמתואר למעלה, השם "המחשבי" של הקובץ (למשל Ran5.wk1).

כל אחד מהקבצים הללו יכיל מקרו שמחזיר את תוכן העניינים של הסדנא:

```
\m   /frMenu~{quit}
```

3. WS_5\DataN [כל מספר = N]

במחיצות אלו ימוקמו רק קובצי ASCII עם נתונים במבנה של מטריצות מספרים, תמליל ותמליל משולב עם מספרים בכל מבנה שנרצה. אין צורך להקים את הקבצים הללו מראש. בהמשך נלמד כיצד להקים קבצים כאלה באופן אוטומטי דרך הלוטוס, כיצד הם נסגרים וכיצד הם מתעדכנים.

## 5.5 הגדרות

לפני שנמשיך, הבה נגדיר מספר מונחים שישמשו אותנו בהמשך.

1. סביבה.

איזור בדיסק המכיל מקבץ של מחיצות וקבצים שהפעלתם נותנת תפוקה ייחודית. למשל,

עיבוד תמלילים      לוטוס 123 גירסה 2.2

2. סביבת יישום לוטוס.

סביבה המופעלת באמצעות Menu.wk1, למשל סדנא 5.

3. יישום לוטוס - כל קובץ wk1.

4. צומת - Menu.wk1 לשליפת קובץ תוכן Menu.wk1 אחר.

5. תוכן עניינים Menu.wk1 להפעלת סביבת יישום לוטוס מסוים.

## 5.6 מבנה מורחב

עתה נוכל לפתח Intra-Macro עד שיהיה מוכן להפעלה שוטפת וללא תקלות. לשם כך נוסיף מרכיבים לכל שעשינו עד כה.

נגדיר תחילה Range Names.

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | | תוכן עניינים | | | \z \0 | {?}/c{bs}{r 2}xData1~ |
| 2 | | /////// | | | | /fr |
| 3 | | | | | xData1 | ← תא ריק |
| 4 | | 1 | סדנא | WS_1 | \a | ~ |
| 5 | | | | | \b | |
| 6 | | 2 | | WS_2 | Msg1 | |

<u>הסבר:</u>

- למקרו 0\ נקרא שם נוסף: z\
- נוסיף עוד שני מקרו: b\ ו-a\
- נוסיף הודעה בשם **Msg1**
- על כל אלה נגדיר עוד שני תחומים:

/rncXX1~b2~ /rncHide1~d1.f60~

[2y- לבסוף נגדיר /ru את התחומים הבאים:

a1.a60 b1 c2.d60 **xData1**

נציג עכשיו את כל המקרו ללא הסברים.

**המקרו השולף**

```
\z  \0 {restart}{windowson}{paneloff clear}{frameoff}{indicate Msg1}{home}{?}
       {windowsoff}/c{bs}{r 2}~xData1~{onerror \z}
       {if @cell("type",xData1)="b"}{branch \z}
       /fr
              <------------------ [תא ריק]
       ~{branch \z}
```

<u>הערה:</u> אני נוטה לכתוב מקרו עד 240 תווים בתא. כאן פיצלתי את פיסקת המקרו על פני מספר שורות לנוחות הקריאה. אפשר לכתוב את כל שלוש השורות הראשונות של המקרו השולף בתא אחד.

כניסה לעמדת מוצא לפני ביצוע /fs

```
\b  {indicate}/rfhHide1~/wgpe/wtc{home}{goto}XX1~/wtb{quit}
```

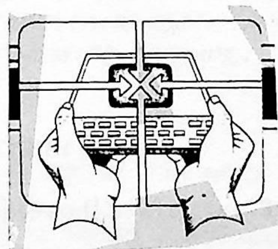
כדי ל"פתוח" את ה-Intra-Macro הזה להכנסת שינויים מבצעים תחילה Ctrl-Break ואחר כך,

```
\a {indicate}/wtc/wgpd/rfrHide1~{quit}
```

ולבסוף, **Msg1** הוא השם המגדיר את התא עם הלוגו. כותבים את התמליל של הלוגו בתא שהוגדר Msg1. המשפט {indicate Msg1} מציב את התמליל הזה בשורה העליונה ביותר של המסך.

קבוצה נוספת של פיסקאות נועדה לאפשר יציאה מתוכן העניינים אל יעדים שונים.

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | | תוכן עניינים | | | \z \0 | {?}/c{bs}{r 2}xData1~ |
| 2 | | /////// | | | | /fr |
| 3 | | | | | xData1 | <— תא ריק |
| 4 | | 1 | סדנא | WS_1 | \a | ~ |
| 5 | | | | | \b | |
| 6 | | 2 | | WS_2 | Msg1 | |
| 7 | | | | | | |
| 8 | | | | | | |
| 9 | | צומת הסדנאות | צא ל | {esc 6}/fdc:\Service\~/frMenu | | |
| 10 | | מיפוי הזיכרון | | {system "4"}  ["4" הוא קובץ bat להפעלת תוכנת שירות] | | |
| 11 | | מערכת ההפעלה | | {esc 6}/s | | |
| 12 | | גיליון חדש | | {esc 6}/wey{quit} | | |
| 13 | | לוטוס | צא מ | {esc 6}/qyy{quit} | | |



"לב" המערכת

# פרק 2

# ניהול תוך גיליוני

## 1. שרשור כתובות

הסמן נמצא בתא כל שהוא. בתא אחר שנקרא **RR1** כתוב e7. רוצים להציב את הסמן בכתובת e7 בעזרת משפט מקרו ש"יאמר" לסמן: "לך והתייצב בכתובת e7, הרשומה במקום שנקרא **RR1**." למעשה, מתואר כן מצב של שרשור כתובות, כמו במשחקים ששיחקנו בצופים: "לך לכתובת **א** ושם יש פתק. על הפתק רשומה כתובת **ב**. תגיד לדודו שילך לכתובת **ב** ויחכה שם".

רבים שואלים את השאלה הזאת. בעזרת **תרגיל** המציג פתרון אפשרי נדגים את החשיבה האינטגרטיבית של Intra-Macro.

נאמר שביצענו ~rncRR1~b5/ וכתבנו e7 בתא **RR1**.

נכתוב עכשיו את משפט המקרו **a**\

```
\a  +"{goto}"&RR1&"~{quit}"
```

התפוקה של **a**\ היא הצבת הסמן בתא e7, או בכל כתובת שתהיה רשומה ב-RR1.

הבה ונאמר כי רוצים:

אם **א** אז יוצב הסמן ב-e8 אבל אם לא **א** אזי יוצב הסמן ב-f5.

ונאמר עוד שביצענו ~rncTest1~b6/ וקבענו כי

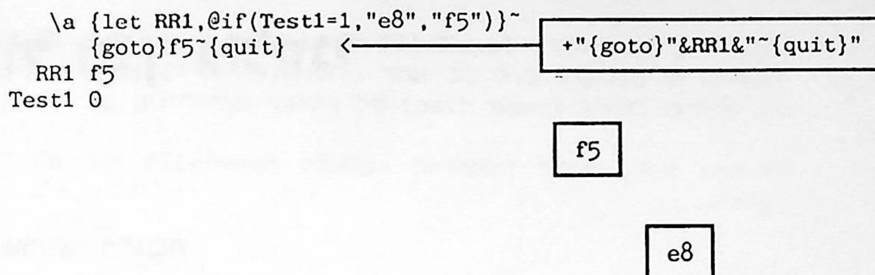
אם **א** אז **Test1**=1, אחרת - **Test1**=0.

עכשיו נכתוב את פיסקת המקרו הבאה, ונקרא לה **a**\:

```
\a  {let RR1,@if(Test1=1,"e8","f5")}~
    +"{goto}"&RR1&"~{quit}"
```

זו הפיסקה שתניב את התפוקה המבוקשת.

פיסקת Intra-Macro שנכתוב לביצוע הפעולה:

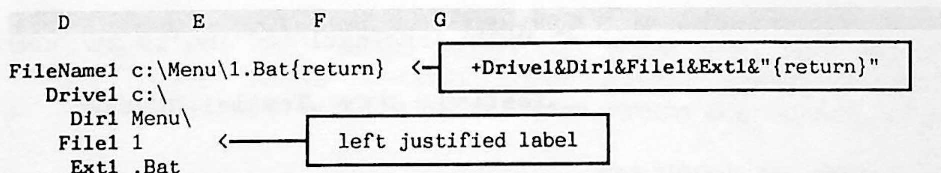



בתרגיל זה גלומים עקרונות רבים. מודולריות, ארגומנטים במקרו, תשתית לאוטומציה, משפטים ופיסקאות במקרו ועוד. בהמשך אדגים דרכי חשיבה שבעזרתם יכול כל משתמש "להמציא פטנטים" רבים שיעזרו לו לקבל את מירב התפוקות מהלוטוס.

## 2. מושגים ראשונים במודולריות

את המושגים הראשונים במודולריות ב-Intra-Macro נבחן ביישום לוטוס שהתפוקה שלו היא קובצי bat. במלים אחרות, נלמד להפעיל את הלוטוס כתוכנת עורך (Editor) לכתיבת קובצי bat שיאוחסנו במחיצה c:\Menu וישמשו לטעינת תוכנות לזיכרון ממערכת ההפעלה.

### 2.1 מודולריות בשם הקובץ




<u>הסבר:</u>

- לפנינו **משפט מקרו** מסוג מיוחד: **משפט שירות במבנה מודולרי.** מודולריות, בצורות שונות, תחזור ותופיע פעמים רבות בהמשך.

במשפט הזה מגדירים חמישה שמות איזורים - Range Names (השמות המודגשים בעמודה D) ע"י rnlr/. שים לב לכך שהשמות צריכים להיות מיושרים לימין (right justified) בתוך התא.

בעמודה E בתא שמול **FileName1** כותבים את המשוואה שבמסגרת. בעמודה זו מכניסים מול כל אחד מהשמות האחרים (ביישור לשמאל) את מרכיבי הנתיב (path name) של הקובץ שמעוניינים בו.

כתוצאה מכך, תחזיר המשוואה בכתובת **FileName1** את שם הקובץ במלואו.

נשנה את תפוקת המשפט הזה ע"י שינוי במרכיביו, למשל:

אם נכתוב \:a בכתובת **Drive1**, במקום \:c שבדוגמה, אזי נקבל מיד ב-**FileName1** את התוצאה

```
a:\Menu\1.Bat{return}
```

אם נכתוב ' (גרש עבור blank) בכתובת **Dir1** וב-**File1** נכתוב 2 (יישור לשמאל - left justified label), נקבל

```
c:\2.Bat{return}
```

<u>תרגיל:</u> הפעל את המחשב ונסה את כל האפשרויות ללא היסוס. ככל שתחוש יותר את אפקט המודולריות בשלב מוקדם ופשוט זה, יהיו הדברים בהמשך קלים ומובנים יותר.

## 2.2 משפט שירות ומשפט מפעיל

ומה פשר המלה {return} שבשורה הראשונה שבמשפט המקרו? **מילת** מקרו זו הופכת את שם התחום הרגיל למשפט שירות במקרו. בשיחה רגילה בין בני אדם משלבים לעתים ביטוי או משפט שלם (**מאמר מוסגר**) שחורג מהקשר הדברים במשפט הנוכחי. בסיומו חוזרים **בדיוק לאותו המקום שבו היינו ממשיכים ללא החריגה אל המשפט המוסגר.** ב-Intra-Macro נכנה את **המשפט הנוכחי** בשם **משפט מפעיל** ואת **המשפט המוסגר** נכנה בשם **משפט שירות.** באנגלית קוראים לפעולת סגירת הסוגריים באמצע משפט המקרו - subroutine call ולמשפט המקרו שבתוך הסוגריים - subroutine.

נבחן למשל את המשפט הבא

```
1_1 {open FileName1,w}
    {close}
    {quit}
```

משפט זה הוא "משפט מפעיל", שכן הוא מתחיל בציון Range Name ומסתיים ב-{quit}. באמצעות משפט זה מבצע המחשב את הפעולות הבאות:

1. פותח קובץ ששמו מופיע בכתובת **FileName1**.
2. לא כותב דבר (גודל הקובץ יהיה 0 בתים).

3. סוגר את הקובץ.
4. אם הקובץ קיים, הוא מבטל את תוכנו ומשאירו כקובץ ריק.

משפט המקרו הזה תמיד מצליח, והוא יחזור ויופיע פעמים רבות בספר זה.

המלה **FileName1** בתוך קטע המשפט {open **FileName1**,w} היא **קריאה למשפט שירות** (כמו פתיחת סוגריים לקראת מאמר מוסגר בשפת דיבור) מכיון שהיא Range Name.

בתא **FileName1** כתוב c:\Menu\1.Bat{return} ומהמאמר המוסגר הזה המחשב "יודע" את שמו המלא של הקובץ המיועד לפתיחה. בסוף המאמר המוסגר הזה כתוב {return} - זאת פעולה של סגירת הסוגריים. המלה {return} "מודיעה" למחשב לחזור ולהמשיך לקרוא את המשפט המפעיל במקום שיצאנו ממנו.

יוצא מכאן, כאילו כתבנו {open "c:\Menu\1.Bat",w}. אלא שהמבנה המודולרי מאפשר לשנות מרכיבים שונים בשם הקובץ ולשנות על ידי כך את תפוקת המשפט המפעיל בלי לכותבו מחדש כל פעם.

שינוי קטן במשפט המפעיל יהפוך אותו ממשפט מפעיל פשוט למשפט שירות:

```
1_1 {open FileName1,w}
    {close}
    {return}
```

כלומר, סיומת המשפט היא {return} במקום {quit}.

נחזור עכשיו למשפט המפעיל הראשי:

```
\p {1_1}/pfX{esc}{FileName1}~
```

המשפט הראשי הזה מפיק תת-מטלה בתוך פיסקת מקרו ששמה p\. קבוצת מלים זו היא משפט שלם: היא מסתיימת בסימן ~ (ENTER), כמו נקודה בסוף משפט, ומכילה את כל המרכיבים של משפט בשפה רגילה (כגון נושא, נשוא, פועל וכו').

משפט זה הוא חלק ממערך מטלות שלם, שתפוקתו היא קובץ ה-ASCII המבוקש. מערך משפטי מקרו אלה הוא **פיסקת המקרו:**

```
\p {1_1}/pfX{esc}{FileName1}~rrMenu1~omnouqgq{quit}
```

המלה **Menu1** מציינת שם-תחום של האיזור אשר בו ייכתב התמליל הנדרש לקובץ.

**<u>תרגיל:</u>** כתוב את המקרו הזה בגיליון ונסה כל שעולה על דעתך.

> **זהירות !!!** המקרו הזה חזק ביותר ועלול למחוק קבצים קיימים. רצוי להקים מחיצה מיוחדת לניסיונות ולהגדיר שמות קבצים שעדיין לא קיימים.

## 2.3 Intra-Macro לעריכת קובצי ASCII

כשמחברים את המרכיבים הללו וכותבים אותם באיזור המקרו (ראה פרק 1) של Intra-Macro זה, מקבלים:

```
       \m /frMenu~{quit}
       \p {1_1}/pfX{esc}{FileName1}~rrMenu1~omnouqgq{quit}
      1_1 {open FileName1,w}
          {close}
          {return}
FileName1 +Drive1&Dir1&File1&Ext1&"{return}"
```

---

```
   Drive1 c:\
     Dir1 Menu\
    File1 1        <------  left justified label
     Ext1 .Bat
```



מבנה הגיליון יהיה כך:

```
A                    B          C            D          E          F
                    \m   /frMenu~{quit}
                    \p   {1 1}/pfX{esc}{FileName1}~rrMenu1~.....
                   1 1   {open FileName1,w}
                         {return}
                         {close}
             FileName1   c:\Menu\1.Bat
------------------------------------------------------------------
                Drive1   c:\
                  Dir1   Menu\
                 File1   1
                  Ext1   .Bat
------------------------------------------------------------------
```

רוחב עמודה A = 240

Menu1 →

```
echo off
cls
cd\123r22
123
cd\
cls
menu
```

הפעלת p\ תיצור batch file בשם c:\Menu\1.Bat שיכלול את כל הטקסט הכתוב בתחום **Menu1**. כתוצאה מכך, כאשר ממסך DOS נקיש 1 ואחר-כך Enter - יעלה הלוטוס.

אם קובץ זה קיים, יוחלף תוכנו בתמליל שב-**Menu1**. אין הגבלה לקובצי bat בלבד. אפשר ליצור בדרך זו כל קובץ ASCII שהוא. השימוש בגיליון לוטוס לכתיבת התמליל מאפשר הפעלת מבחר גדול של פקודות לעריכה.

ראינו עד כה כי Intra-Macro זו מערכת כללים לדו-שיח עם המחשב הדומה מאוד לכללי היסוד של כל שפת דיבור אנושית רגילה. דו-שיח עם המחשב מתנהל במבנה של אותיות ומלים המתחברות יחד למשפטים ולפיסקאות. פיסקאות מקרו אלו, כשהן שזורות זו בזו ובמרכיבי הגיליון האחרים, גורמות למחשב ליצור את התפוקות המבוקשות.

במערכת כללים זאת למדנו:

- מודולריות בשם הקובץ,
- משפט שירות,
- משפט מפעיל פשוט,
- משפט מפעיל מורכב,
- פיסקה.

## 3. מושגים מתקדמים במודולריות - שילוב /fit עם Data Parse

המודולריות ב-Intra-Macro באה לידי ביטוי מוגבר כשמשלבים במשפטי המקרו פונקציות והודעות. כך מקבלים תפוקת יישום מגוונת ובו-זמנית נוחות הפעלה מירבית למפעיל. הדוגמא הנדושה ביותר, מן הסתם, בהוראת מחשבים היא דוגמת הקמה של ספר טלפונים. גם אנו נציג את מהלך העבודה להקמת ספר טלפונים.

הפיתוח בהמשך פרק זה יתמקד במרכיבים להבנת עקרונות. לפיכך לא יהיה זה "היישום הטוב ביותר האפשרי". דברים רבים יישארו "פתוחים". חלקם יסגרו בפרקים הבאים וייתרם יפתור התלמיד היצירתי בעצמו. בפרקים הבאים יוצגו דוגמאות רבות ומגוונות.

### 3.1 שלב א: תפוקות ומבנה הקבצים

שם היישום: ספר טלפונים

**הגדרת תפוקות**

1. קליטת שמות ומספרי טלפון בשלושה איזורים:

   א. ישראל.
   ב. ניו-יורק.
   ג. וושינגטון.

2. שליפת כל השמות מאיזור נתון.

**הגדרת מבנה התפעול**

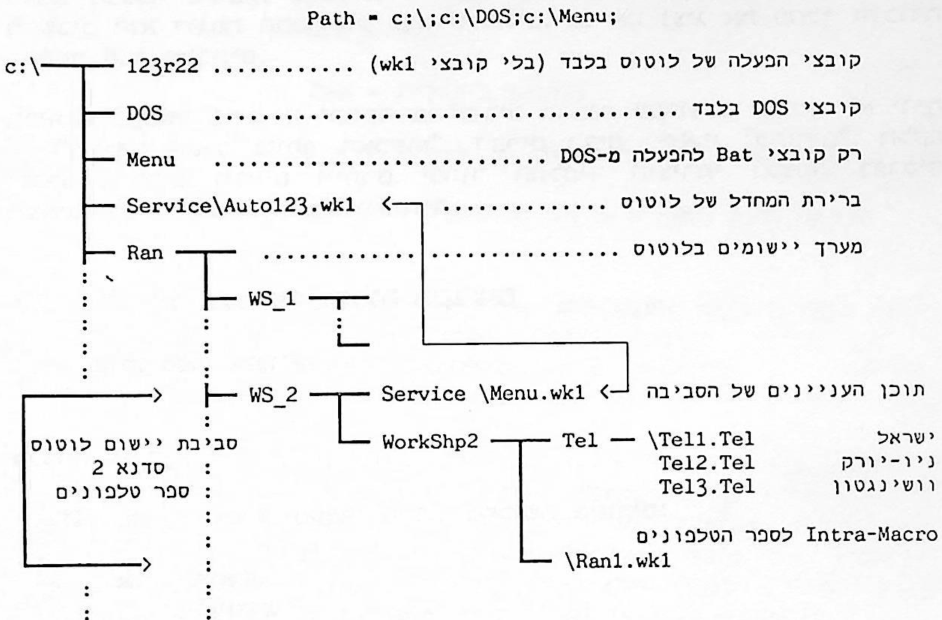
הנתונים יישמרו בקובצי ASCII והפעלת היישום תתבצע באמצעות Intra-Macro המוגדר בקובץ wk1.

למבנה זה יתרונות רבים על פני שמירת נתונים בקובצי wk1. חלק מהיתרונות הללו יתגלו בהמשך וחלק אציין עכשיו.

1. חיסכון במקום בדיסק.
2. מהירות שליפה גבוהה.
3. התפעול מתבצע באמצעות קובצי wk1 קטנים במהירות גבוהה.
4. גמישות רבה במיקום מאגר הנתונים בשלמותו, או פיזורו בחלקים במרחבי הדיסק.
5. גישה קלה ונוחה לנתונים מקובצי wk1 שונים.

## 3.2 שלב ב: מיפוי עבור Meta-Macro

(הקמת סביבת היישום)




בעת מיפוי הזיכרון עבור Meta-Macro מחליטים על מיקום סביבת העבודה ומרכיביה. כאן ממוקם ספר הטלפונים במערך סדנא מספר 2. בתוך המחיצה WS_2 נפתחות שתי מחיצות (sub-directories):

Service: במחיצה יהיה קובץ Menu.wk1 המותאם לסביבה זו. קובץ זה קשור אל צומת המערכת מצד אחד ומקשר בין מרכיבי סדנא מספר 2 מצד שני.

WorkShp2: במחיצה זו יהיו שני מרכיבים:

1. קובץ Ran.wk1 שהוא גיליון Intra-Macro בלוטוס.
2. מחיצה בשם Tel ובתוכה שלושה קובצי ASCII:
   - Tel1.Tel - נתונים בישראל.
   - Tel2.Tel - נתונים בניו-יורק.
   - Tel3.Tel - נתונים בוושינגטון.

לשמות קובצי הנתונים הללו חשיבות רבה באוטומציה של הפעלת המערכת. כל שם קובץ המחזיק נתונים מורכב משם (Tel) וממספר. כלומר, הצורה הכללית של שם קובץ הנתונים תהיה: Teln.Tel [n = כל מספר].

לאמיתו של דבר, אני נוהג להשתמש בצורה הכללית הבאה:

Xxx_n.Ran

כאשר,

Xxx = אותיות באנגלית

n = 1..9999 [כל מספר מ-1 ועד 9999]

הסיומת הינה זיהוי אישי של מי שאליו צריך לפנות לבירורים.

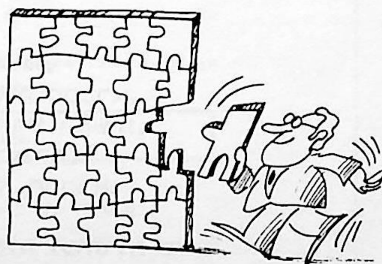
שילוב מספרים בשם הקובץ יוצר מערך סימול לאוטומציה של שמירת ושליפת נתונים. בתרגיל הנוכחי למשל, הערך 1 תמיד יהיה ספר הטלפונים בישראל, 2 - בניו-יורק, 3 - בוושינגטון. השימוש בסמלים הללו והקשר שהם מאפשרים בין המשתמש והנתונים יודגם בהמשך הדברים.

מיפוי הסביבה זו מטלה שמתבצעת תוך דקות ספורות. ואחרי שהדבר נעשה אין יותר צורך להיכנס לתוכה. הסביבה מוקמת עד רמת המחיצות **בלבד**. את קובץ תוכן העניינים מעתיקים ממקום אחר ומכניסים בו את השינויים הדרושים. את קובץ ה-Intra-Macro בלוטוס מקימים כפי שיוסבר להלן. קובצי הנתונים לצורך תרגיל זה הוכנו ידנית, אך במהלך הפרקים הבאים נלמד כיצד לגרום לפתיחה (או ביטול) אוטומטי של קבצים, מבלי שהמפעיל יהיה בכלל מודע לכך.

ועתה נעלה למסך גיליון לוטוס חדש ונבצע:

/fdc:\Ran\WS_2\WorkShp2\~ /fsRan1~

ובסיום יהיה השטח מוכן להקמת ה-Intra-Macro.



## 3.3 שלב ג: מיפוי עבור Intra-Macro

### איזור הפעילות הגלויה

בפרק 1 התייחסתי לארבעה איזורי פעילות בגיליון בהם מארגנים את ה-Intra-Macro. הראשון שבהם הוא איזור הפעילות הגלויה. גודלו של איזור זה איננו מותנה בגודל המסך מכיוון שאפשר "לגלול" את המסך על פני שורות רבות ואפשר ל"הקפיצו" למקומות שונים בגיליון. רצוי למקם את האיזור הזה בפינה השמאלית העליונה של הגיליון. כאן הוא המסך שדרכו מתבצעת האינטראקציה בין האדם והמחשב ולכן חשיבותו המרובה.

איזור האינטראקציה הזאת (הפעילות הגלויה) מכיל מספר "משטחים":

1. מקרא.
2. הודעות מהמחשב.
3. כותרות.
4. קלט/פלט.

אין אלה "כללי ברזל" שאין לעבור עליהם אבל הם קיימים במרבית, אם לא בכל, יישומי לוטוס מתקדמים.

והרי דוגמה:

```
                  A                           B          C        D          E
                                                  MBA LAC רן אברהמי 1990 (c)
1
2    Alt m =  לתוכן                            Alt b = מחק הכל
3                          מדריך טלפונים
4
5    Alt i =  ישראל     Alt n = New York     Alt w = Washington
6
7                    Name                                 Telephone
8    ======================================== ====================
9                 Last                   First      Area     Number
10   ========================= ============= ====== ============
11
12   ////////////////////////////
```

<u>הסבר:</u>

בשורה הראשונה: לוגו.
בשורה מספר 2: חלק ממקרא.
שורה מספר 3: כותרת ראשית.
שורה 4: ריקה (לכאורה).
שורה 5: שאר חלקי המקרא.
שורה 6: ריקה.
שורות 7..10: כותרות וכותרות משנה.
שורה 11: ריקה.
שורה 12: ריקה, סמן בעמודה A.

בנוסף לכל אלה הוגדרו שני תחומים נוספים כדלקמן,

/rncXX11~a4~ /rncData1~a12.d12~

משמעות כל אלה תתברר תוך כדי בחינת המקרו.

## איזור הפעילות הסמויה

איזור זה הוא "משטח" עבודה המשתרע על פני עמודה אחת או יותר שבו מתבצעת ה"עבודה" בעת הפעלת היישום. אחרי שהיישום נכתב סופית רצוי להקפיא את המסך על איזור הפעילות הגלויה {windowsoff}
{paneloff clear} בעת ביצוע משפטי המקרו. הקפאה זו מונעת התרוצצות על פני המסך ומגבירה את מהירות הביצוע.

בדוגמה הנוכחית נקבעה עמודה F לאיזור הפעילות הסמויה ובה הוגדרו שני תחומים,

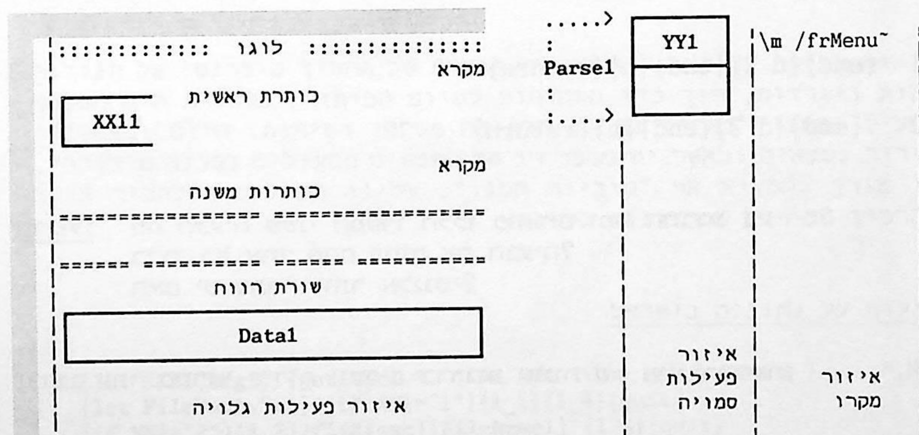
/rncYY1~f1~ /rncParse1~f1.f2~

## איזור המקרו

עמודות G ו-H נבחרו למיקום משפטי המקרו: עמודה G לשמות ועמודה H לפיסקאות.

## מפת Intra-Macro

כך נראית מפה של גיליון Intra-Macro זה:



## 3.4 שלב ד: כתיבת מקרו

**נוהלי שליפה ושמירה**

נאמר שקיים קובץ נתונים מעורב - טקסט ומספרים - במבנה ASCII. ועוד נאמר שהקובץ ערוך במבנה של Data Base כלומר, שדות ורשומות כאשר כל שדה מכיל את הנתונים של עמודה אחת תחת כותרות המשנה באיזור הפעילות הגלויה.

התפוקה הנדרשת:

1. לייבא את הנתונים ולהציב כל שדה בעמודתו כאשר טקסט מופיע כ-label ומספר כ-value.
   סדר הפעולות:

   א. לייבא את הקובץ אל תא YY1.
   ב. לבצע Data Parse כאשר ~dpoData1/

2. לשמור נתונים איזוריים במבנה ASCII.

3. להקים תשתית משפטי שירות.

מתי מקימים משפט שירות? כל אימת שמשפט מקרו מופיע יותר מפעם אחת בפיסקה אחת או יותר, רצוי להופכו למשפט שירות. הניסיון מראה, שאם משתמשים ב"מודול" אחד פעמיים, יש סיכוי רב שנשתמש בו לפחות פעם אחת נוספת. יש רמות שונות של משפטי שירות, החל ממשפט שירות שישתלב כמעט בכל משפט מקרו אחר וכמעט בכל יישום לוטוס, וכלה במשפט שירות ייחודי לפיסקה אחת ביישום אחד.

הבה נקים שני משפטי שירות אוניברסלים. משפטים אלה לא יוסברו, אך משמעותם תתבהר תוך כדי שילובם במשפטים ברמה גבוהה יותר.

```
31   {end}{d 2}{end}{u}{return}

32   .{end}{d 2}{end}{u}{return}
```

__תרגיל:__ מה הבעיה ששני המקרו הללו פותרים ומה ההבדל ביניהם?
כיצד כל אחד מהם פותר את הבעיה?
האם יש פתרון יותר אלגנטי?

נכתוב עתה משפטי שירות נוספים בדרגות שונות של אוניברסליות:

```
\b       {blank Data1}{blank XX11}~/wtc{home}{goto}Data1~/wth{return}
1_1      {open FileName1,w}
         {close}
         {return}
1_2      {paneloff}{windowsoff}/wtc{goto}YY1~{return}
1_3      /rncParse1~{bs}{u}.{31}~
         /dpiParse1~oData1~g
         /re{31}~{home}{goto}Data1~/wth{windowson}{panelon}{return}
1_4      /rncData1~.{31}~/pfX{esc}{FileName1}~rrData1~omnouqgq{return}
FileName1 +"c:\Ran\WS_2\WorkShp2\Tel\Tel"&FileNum1&".Tel{return}"
```

נגדיר מספר תחומים אשר כולם יהיו באיזור המקרו:

```
NN1        <———————— 1 = שמור   2 = שלוף
FileNum1   <———————— 1 = ישראל   2 = ניו-יורק   3 = וושינגטון
Msg1       * * *   בישראל   * * *
Msg2       * * *   בניו-יורק   * * *
Msg3       * * *   בוושינגטון   * * *
Msg4       לא אירע כלום כי לא הקשת 1 או 2
```

נותר לנו רק להגדיר parse line בתא YY1, כך שיתאים למבנה השדות והגדרתם (טקסט או מספר) בהתאמה להגדרות בכותרות המשנה:

```
   שם משפחה              שם פרטי        איזור    מספר טל
L>>>>>>>>>>>>>>>>>>>>>>>>>>>>L>>>>>>>>>>>>>>>>>L>>>>>>L>>>>>>>>>>>>>>>
```

<u>**תרגיל:**</u> הגדר במלים את מטרתו של כל אחד ממשפטי השירות שלמעלה.

## 4. משפטים מפעילים

התיאור שניתן עד כאן הוא למעשה תולדה של תהליך קיברנטי של תנודות עולות ויורדות, תוך כדי התקדמות לסיום הפיתוח. בראשית היה המשפט המפעיל וממנו "צמחו" במהלך הכתיבה משפטי השירות. היום, כשמשפטי השירות נמצאים וכאשר יש מספר רב של משפטים מפעילים במבנה מודולרי, אין צורך להמציא את "קוביות הלגו". עלינו רק ללמוד להכיר אותן ולהרכיב מהן כל שברצוננו לעשות.

<u>פעולות על נתונים בישראל</u>

```
\i {let XX11,Msg1}~{getlabel " ... שמור או שלוף ... ( 1 או 2 ) ... ",NN1}~
   {let FileNum1,"1"}~{if NN1="1"}{1_1}{1_4}{quit}
   {if NN1="2"}{1_2}/fitX{esc}{FileName1}~{1_3}{quit}
   {let XX11,Msg4}~{quit}
```

## פעולות על נתונים בניו-יורק

```
\n {let XX11,Msg2}~{getlabel " שמור או שלוף ... ( 1 או 2 ) ... ",NN1}~
   {let FileNum1,"2"}~{if NN1="1"}{1_1}{1_4}{quit}
   {if NN1="2"}{1_2}/fitX{esc}{FileName1}~{1_3}{quit}
   {let XX11,Msg4}~{quit}
```

## פעולות על נתונים בוושינגטון

```
\w {let XX11,Msg3}~{getlabel " שמור או שלוף ... ( 1 או 2 ) ... ",NN1}~
   {let FileNum1,"3"}~{if NN1="1"}{1_1}{1_4}{quit}
   {if NN1="2"}{1_2}/fitX{esc}{FileName1}~{1_3}{quit}
   {let XX11,Msg4}~{quit}
```

במפת ה-Intra-Macro ייראה איזור המקרו כך:

```
          \m /frMenu~{quit}
          \b {blank Data1}{blank XX11}~/wtc{home}{goto}Data1~/wth{return}
          \i {let XX11,Msg1}~{getlabel " שמור או שלוף ... (1 או 2)...",NN1}~
             {let FileNum1,"1"}~{if NN1="1"}{1_1}{1_4}{quit}
             {if NN1="2"}{1_2}/fitX{esc}{FileName1}~{1_3}{quit}
             {let XX11,Msg4}~{quit}
          \n {let XX11,Msg2}~{getlabel " שמור או שלוף ... (1 או 2)...",NN1}~
             {let FileNum1,"2"}~{if NN1="1"}{1_1}{1_4}{quit}
             {if NN1="2"}{1_2}/fitX{esc}{FileName1}~{1_3}{quit}
             {let XX11,Msg4}~{quit}
          \w {let XX11,Msg3}~{getlabel " שמור או שלוף ... (1 או 2)...",NN1}~
             {let FileNum1,"3"}~{if NN1="1"}{1_1}{1_4}{quit}
             {if NN1="2"}{1_2}/fitX{esc}{FileName1}~{1_3}{quit}
             {let XX11,Msg4}~{quit}
         1_1 {open FileName1,w}
             {close}
             {return}
         1_2 {paneloff}{windowsoff}/wtc{goto}YY1~{return}
         1_3 /rncParse1~{bs}{u}.{31}~
             /dpiParse1~oData1~g
             /re{31}~{home}{goto}Data1~/wth{windowson}{panelon}{return}
         1_4 /rncData1~.{31}~/pfX{esc}{FileName1}~rrData1~omnouqgq{return}
          31 {end}{d 2}{end}{u}{return}
   FileName1 +"c:\Ran\WS_2\WorkShp2\Tel\Tel"&FileNum1&".Tel{return}"
         NN1     <------------ 1 = שמור    2 = שלוף
   FileNum1      <-------- 1 = ישראל   2 = ניו-יורק   3 = וושינגטון
        Msg1            * * *     בישראל     * * *
        Msg2            * * *    בניו-יורק   * * *
        Msg3            * * *    בוושינגטון  * * *
        Msg4          לא אירע כלום כי לה הקשת 1 או 2
```

**תרגילים:**

1. הסבר במלים את מטרתו של כל אחד מהמשפטים המפעילים ואת סדר הפעולות שלו.

2. האם נראית סיבה לסידור המשפטים מלמעלה למטה? אם כן - מה היא?

3. שלוף משפטי שירות נוספים מהמשפטים המפעילים וכתוב אותם.

4. בפרק הבא נלמד כיצד לכתוב את כל שלושת המשפטים המפעילים שלמעלה **במשפט מפעיל אחד.** נסה לבצע זאת בעצמך כבר עכשיו.

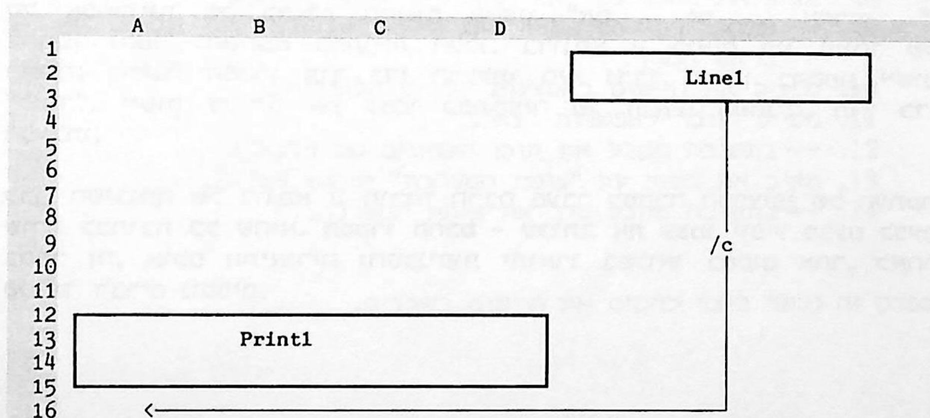
## 5. מושגים ראשונים בארגומנטים

הולכת הסמן ממקום למקום על פני הגיליון היא תפוקה מבוקשת בפיסקאות מקרו רבות ומסיבות שונות. הבה נבחן את המצב הבא.

נתון תחום בשם **Print1** המשתרע על פני שטח כל שהוא המכיל נתונים ונוסחאות. הפינה השמאלית העליונה של שטח זה היא a12. נאמר עוד, שיש במקום כל שהוא בגיליון שורה של נוסחאות בשם **Line1** כדלקמן,

| E | F | G | H |
|---|---|---|---|
| 0 | 0 | 0 | @sum(e1.g1) |

התפוקה המבוקשת היא להעתיק את **Line1** לעמודה a, שורה אחת אחרי השורה האחרונה של **Print1**, כאשר השורה האחרונה של **Print1** אינה קבועה מראש, אלא משתנה מפעם לפעם.



פיסקת ה-Intra-Macro לביצוע המטלה הזאת היא:

```
 /a {1_2}/cLine1~{Adr1}~{quit}
 1_2 {let Adr1,+"a"&@string(@rows(Print1)+12,0)&"{return}"{return}
Adr1
```

פיסקה זו מאפשרת להוסיף שורה חדשה לבסיס הנתונים, להגדיר מחדש את התחום שכולל את הבסיס הזה ולהוסיף אחר כך שורה אחת, או מספר שורות. בפיסקה זו מודגמים מספר עקרונות מתקדמים בכתיבת Intra-Macro.

פירוק הנוסחה:

```
+"a"&@string(@rows(Print1)+12,0)&"{return}"
```

1. מספר השורות של **Print1** בדוגמה זו — `@rows(Print1) —> 4`

2. מספר השורה שאחרי השורה האחרונה של **Print1** — `@rows(Print1)+12 —> 16`

3. כתובת התא המבוקש — `+"a"&@string(@rows(Print1)+12,0) —> a16`

4. `+"a"&@string(@rows(Print1)+12,0)&"{return}" —> a16{return} = Adr1`

את הנוסחה המורכבת הזאת כותבים כארגומנט במקרו { ... ,**let Adr1**}. ההפעלה מפיקה משפט שירות בכתובת **Adr1** המופעל על ידי המשפט הראשי להעתקת **Line1**.

כאן רואים בבירור כיצד שילוב של פונקציות כארגומנטים במקרו מאפשר אוטומציה בהפעלת הגיליון. מה שמתרחש למעשה, הוא כתיבת מקרו על ידי המקרו, תוך כדי ביצוע.

אחד ממקורות כוחו של הלוטוס כשפה לדו-שיח עם המחשב הוא המבחר הרב של אפשרויות של כתיבת התניות מהסוג "אם ... אז ..., אחרת ...". בהמשך נלמד להשתמש בהתניות הללו. בדוגמא זו רואים סוג מיוחד של התניה: כתיבת המקרו תוך כדי הפעלתו היא נדבך ראשון בהקמת יישום "לומד". יישום שכותב את עצמו כפונקציה של תנאים משתנים, תוך כדי הפעלתו.

לכן, חשיבותה של דוגמא זו חורגת הרבה מעבר למקרה הספציפי של העתקת שורה לכתובת כל שהיא. **המקרו החכם - שכותב את עצמו** יופיע מפעם לפעם בספר זה, אולם התיאוריה והטכניקות יתוארו במלואן במקום אחר, לאחר שנלמד דברים נוספים.

# פרק 3

# ניהול תוך גיליוני - תפריט פקודות

## 1. שלבים בהקמת יישום

בפרק 1 הקמנו את התשתית הארגונית לניהול הדיסק ויישומי הלוטוס כהכנה ל-**Meta-Macro**. במיוחד התמקדנו בתוכן העניינים המקשר אותנו עם הדיסק. הקו המנחה הוא אוטומציה מירבית מרגע פתיחת המחשב ועד לסגירתו. בפרק 2 הכרנו מעט את ה-**Intra-Macro** ולמדנו לארגן את הגיליון ולכתוב במבנה מודולרי אינטגרטיבי.

בפרק הנוכחי נסגור את מעגל האוטומציה ונלמד כיצד לנהל את הפעילות התוך גיליונית באמצעות תפריט פקודות. סדר הפעולות יהיה כדלקמן:

1. מפעיל את המחשב.
2. ←רואה על המסך את תוכן העניינים ואת ה-prompt <\:c
3. מקיש 1 ו-ENTER.
4. ←רואה את מסך צומת המערכת ליישומי לוטוס.
5. מציב את הסמן על "סדנא 3 - ניהול היישום ע"י תפריט פקודות".
6. מקיש ENTER.
7. ←בתשובה מקבל את תוכן העניינים של סדנא 3.
8. מציב את הסמן על "ספר טלפונים" והקש ENTER.
9. ←מקבל מסך Intra-Macro לספר טלפונים, עם תפריט פקודות בראש המסך.
10. מפעיל את היישום באמצעות תפריט הפקודות.
11. מסיים ועובר לאפשרות "צא".
12. ←בתשובה מקבל את תוכן העניינים של סדנא 3.
13. מציב את הסמן על "צומת המערכת" ומקיש ENTER.
14. ←בתשובה מקבל שוב את צומת המערכת.

בפרק זה נלמד כיצד להקים את היישום בשלבים.

## 2. שלב א: תפוקות ומבנה הקבצים

שם היישום: ספר טלפונים

**הגדרת תפוקות**

1. קליטת שמות ומספרי טלפון בשלושה איזורים.

   1. ישראל
   2. ניו-יורק
   3. וושינגטון

2. שליפת כל השמות מאיזור נתון.

3. פקודות למחשב.

   1. שלוף - שלוף ספר הטלפונים של איזור מסוים מהדיסק אל המסך.
   2. חפש - חפש מספר טלפון לפי שם משפחה ושם פרטי.
   3. מיין - מיין את הנתונים לפי סדר הא"ב.
   4. שורה - פתח שורה נוספת לקליטת שם נוסף.
   5. הדפס - הדפס את הנתונים שנשלפו.
   6. מסך - מסך חדש ללא כל פעולה נוספת.
   7. צא - יציאה לתוכן העניינים ללא כל פעולה נוספת.
   8. שמור - שמירת ספר הטלפונים של איזור מהמסך אל הדיסק.

4. נוהל תפעול

   מעגל סגור - { ... menucall} <—> ri/

**הגדרת מבנה היישום**

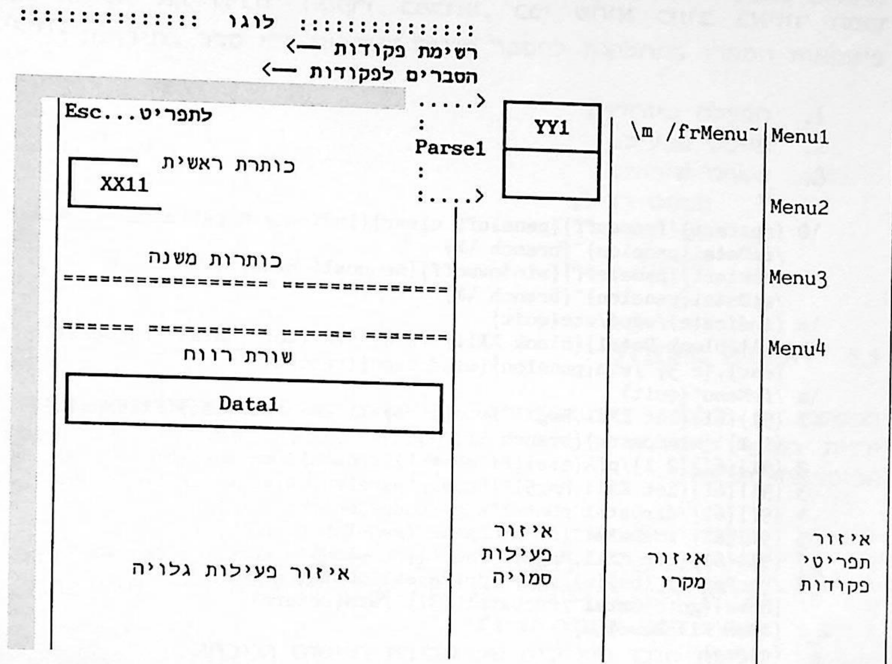
- הנתונים יישמרו במבנה ASCII.
- ההפעלה תתבצע ע"י Intra–Macro בקובץ wk1.
- כל הפעילות התוך גיליונית תתבצע באמצעות תפריט פקודות.

## 3. שלב ב: מיפוי עבור Meta–Macro

**(הקמת סביבת היישום)**

<u>תרגיל:</u> בחן את ה-Intra-Macro שבהמשך פרק זה ושחזר ממנו את המיפוי ל-Meta-Macro לפי הדוגמא שבפרק 2.

## 4. שלב ג: מפת Intra-Macro




מפת ה-Intra-Macro בפרק זה זהה לזו שבפרק 2, פרט לכך שהלוגו מוצג בשורת הדו-שיח הראשונה (מלמעלה), ובמקום. מקרא ל"אלטים" (alt's) יש תפריט פקודות בשתי שורות הדו-שיח הנותרות.

כמו כן, במפה זו נוסף איזור תפריטי פקודות הממוקם מימין לאיזור המקרו.

## 5. שלב ד: כתיבת המקרו

נראה תחילה את חיבור המקרו במלואו, כפי שהוא כתוב באיזור המקרו בגיליון:

```
       \0 {restart}{frameoff}{paneloff clear}{indicate Msg4}{menucall Menu1}
          /riData1{panelon}~{branch \k}
       \k {restart}{paneloff}{windowsoff}{menucall Menu1}{windowson}
          /riData1{panelon}~{branch \k}
       \a {indicate}/wgpd/wtc{quit}
       \b {61}{blank Data1}{blank XX11}~/wtc{home}{goto}Data1~/rncData1~
          {esc}.{r 3}~/wth{panelon}{windowson}{return}
       \m /frMenu~{quit}
        1 {91}{61}{let XX11,MsgX}~/wtc{goto}YY1~{d}/fitX{esc}{FileName1}~
          {1_2}~{windowson}{branch 41}
        2 {91}{61}{2_1}/pfX{esc}{FileName1}~rrData1~omnouqgq{branch \k}
        3 {91}{61}{let XX11,Msg5}~{Beep1}{panelon}{windowson}{branch 41}
        4 {91}{61}/dsrData1~pData1~a~g{windowson}{branch 41}
        5 {91}{61}/rncData1~{d}~/ruData1~{panelon}{branch 41}
        6 {91}{61}{let XX11,Msg5}~{Beep1}{panelon}{windowson}{branch 41}
      1_2 /rncParse1~{bs}{u}.{31}~/dpiPares1~oData1~g/re{31}~
          {home}{goto}Data1~/rncData1~{31}~/wth{return}
      2_1 {open FileName1,w}
          {close}
          {return}
       31 {end}{d 2}{end}{u}{return}
       41 {restart}{paneloff}{windowson}/riData1{panelon}~{branch \k}
       61 {paneloff}{windowsoff}{return}
       91 {onerror b 1,XX11}{return}
      b_1 {61}/wtc{home}{goto}Data1~/wth{panelon}{windowson}{Beep1}
          {branch 41}
    Beep1 {beep 2}{beep 1}{beep 2}{beep 1}{return}
   Drive1 c:
     Dir1 \Ran\WS_3\WorkShp3
  SubDir1 \Tel
    File1 \Tel
      NN1   <-----------  | 1 = ישראל    2 = ניו-יורק    3 = וושינגטון |
     Ext1 .Tel
FileName1 +Drive1&Dir1&SubDir1&File1&NN1&Ext1&"{return}"
     MsgX @@(+"Msg"&NN1)
     Msg1                       בישראל
     Msg2                       בניו-יורק
     Msg3                       בוושינגטון
     Msg4                                               (c) 1990 רן אברהמי
     Msg5              נסה לכתוב בעצמך את המקרו הזה
```

חיבור זה מתחיל בשורה הראשונה מלמעלה באיזור המקרו ונמשך ברצף וללא שורות רווח. שמות הפיסקאות נמצאים בעמודה אחת והפיסקאות עצמן נמצאות בעמודה הימנית הסמוכה. מומלץ למלא כל תא כמעט בכל 240 התווים שאפשר לכתוב בו.

פיסקאות המקרו מתחלקות למספר סוגים עיקריים לפי סדר כתיבתם:

1. הפעלה מיוחדת.
2. משפט מפעיל.
3. משפט שירות.
   1. משפט רגיל.
   2. שם קובץ.
   3. הודעה.

## 5.1 הפעלה מיוחדת

בקבוצה זו נכללות פיסקאות מקרו שנועדו לתת למשתמש אופציה להפעלה ידנית בנוסף להפעלה אוטומטית באמצעות תפריט הפקודות. ביניהן נמצאות הפיסקאות הבאות:

\0 הפעלה אוטומטית של הקובץ מיד עם עליתו למסך.
\a שיחרור הקובץ במטרה לאפשר הכנסת שינויים בכל מקום בגיליון.
\b סגירת הקובץ למצב שבו הוא מוכן להפעלה לפני \0 (בדרך כלל מופעל לפני שמבצעים /fs).
\m שליפת תוכן העניינים של סביבת היישום הנוכחי.

ברוב המקרים יקרא המקרו \0 בשם נוסף - \z - כדי להקל על הפעלתו הידנית. ביישום הנוכחי יש מקרו נפרד לשם כך - \k

```
\0 {restart}{frameoff}{paneloff clear}{indicate Msg4}{menucall Menu1}
   /riData1{panelon}~{branch \k}
\k {restart}{paneloff}{windowsoff}{menucall Menu1}{windowson}
   /riData1{panelon}~{branch \k}
\a {indicate}/wgpd/wtc{quit}
\b {61}{blank Data1}{blank XX11}~/wtc{home}{goto}Data1~/rncData1~
   {esc}.{r 3}~/wth{panelon}{windowson}{return}
\m /frMenu~{quit}
```

## 5.2 משפטים מפעילים

משפטים (פיסקאות) מפעילים מבצעים את המטלות שהוגדרו בתפריט הפקודות. משפט מפעיל תמיד מתחיל ב- Range Name ומסתיים ב-{[1 Kquit} או ב-{branch}.

בקובץ הנוכחי מבצעות הפיסקאות האלו את הפעולות הבאות:

1. שלוף - ישלפו הנתונים מאיזור כל שהוא.
2. שמור - שמירת הנתונים על הדיסק במבנה ASCII.
3. חפש - חפש ומצא רשומה לפי שם משפחה ושם פרטי.
4. מיין - מיון הנתונים שנשלפו לפי סדר הא"ב.
5. שורה - פתיחת שורה חדשה לקליטת רשומה נוספת.
6. הדפס - הדפסת הנתונים.

שתי פיסקאות המוגדרות "מיוחדות" משמשות גם בהפעלה השוטפת באמצעות תפריט הפקודות:

\m צא - יציאה לתוכן העניינים ללא כל פעולה נוספת.
\b מסך - קבלת מסך חדש ללא כל פעולה נוספת.

המקרו הבאים, המהווים את עיקר היישום, עונים על הגדרת התפוקות שלמעלה.

```
1 {91}{61}{let XX11,MsgX}~/wtc{goto}YY1~{d}/fitX{esc}{FileName1}~
  {1_2}~{windowson}{branch 41}
2 {91}{61}{2_1}/pfX{esc}{FileName1}~rrData1~omnouqgq{branch \k}
3 {91}{61}{let XX11,Msg5}~{Beep1}{panelon}{windowson}{branch 41}
4 {91}{61}/dsrData1~pData1~a~g{windowson}{branch 41}
5 {91}{61}/rncData1~{d}~/ruData1~{panelon}{branch 41}
6 {91}{61}{let XX11,Msg5}~{Beep1}{panelon}{windowson}{branch 41}
```

## 5.3 משפטי שירות

משפט שירות הוא משפט מקרו המשמש "מאמר מוסגר" למספר משפטים מפעילים. פרט למשפטי שירות מסוג Msg יסתיים כל משפט שירות במלה {return}. בנוסף להבדלים בין סוגי משפטי השירות לפי המטלות שהם מבצעים נבדלים המשפטים הללו לפי רמת האוניברסליות שלהם.

1. **מקרו שירות לקובץ הנוכחי**

משפט מקרו המיועד לתת שירות למקרו בקובץ זה בלבד - רמת האוניברסליות נמוכה ביותר.

משפט 1_2 ביצוע data parse.

משפט 2_1 הכנה לשמירת נתונים ב-ASCII.

```
1_2 /rncParse1~{bs}{u}.{31}~/dpiPares1~oData1~g/re{31}~
    {home}{goto}Data1~/rncData1~{31}~/ruData1~/wth{return}
2_1 {open FileName1,w}
    {close}
    {return}
```

## 2. מקרו שירות אוניברסליים

ככל שגדל מספר המשפטים "המקבלים שירות" ממשפט מקרו מסוים, עולה רמת האוניברסליות של אותו משפט. למשל, המשפט **{31}** הוא ברמת אוניברסליות גבוהה ביותר מכיון שהוא מופיע כ"מאמר מוסגר" במספר רב מאוד של משפטים אחרים. מהמשפטים הללו רצוי להקים ספריית מקרו. ספרייה זו יכולה להיות בקובץ wk1 שממנו יישלפו המשפטים על ידי fcc/ אל Intra-Macro שנמצא בעריכה. בשלבים מתקדמים יותר, שמים את המקרו האלה בקובץ mlb וטוענים אותו אוטומטית בכל פעם (הסברים נוספים בהמשך).

```
   31 {end}{d 2}{end}{u}{return}
   41 {restart}{paneloff}{windowson}/riData1{panelon}~{branch \k}
   61 {paneloff}{windowsoff}{return}
   91 {onerror b_1,XX11}{return}
  b_1 {61}/wtc{home}{goto}Data1~/wth{panelon}{windowson}{Beep1}
      {branch 41}
Beep1 {beep 2}{beep 1}{beep 2}{beep 1}{return}
```

## 3. שמות קבצים

**שם הקובץ הוא הקשר בין הגיליון לבין הדיסק.** הוא הכתובת בה מוחסנים נתונים והוראות וממנה הם נשלפים. האינטגרציה במערכות Meta-Macro ו- Intra-Macro מבוססת במידה רבה על שמות קבצים במבנה ובאופן המאפשרים אוטומציה מירבית של ניהול הקבצים.

שם הקובץ ב-Intra-Macro הוא סוג מיוחד של משפט שירות. משפט זה הוא פונקציה מורכבת המסתיימת ב-{return}. באמצעותה ניתן להפיק ממשפט מפעיל קצר אחד מספר תפוקות אשר בדרך כלל נדרשים עבורן משפטי מקרו רבים ומורכבים.

```
   Drive1 c:
     Dir1 \Ran\WS_3\WorkShp3
  SubDir1 \Tel
    File1 \Tel
      NN1  <----  | 1 = ישראל   2 = ניו-יורק   3 = וושינגטון |
     Ext1 .Tel
FileName1 +Drive1&Dir1&SubDir1&File1&NN1&Ext1&"{return}"
```

### 4. הודעות

הפעלת קובץ יוצרת מצב של דו-שיח (אינטראקציה אקטיבית) בין האדם לבין המחשב. על ידי הקרנת הודעות לפי הצורך במקום בולט שיועד לכך מראש מקבל המפעיל מידע מהמחשב בתגובה על פעולותיו. הודעות אלו יכולות לשמש למטרות שונות, החל מהקרנת לוגו וכלה בהודעה המשולבת ביישום לומדה.

```
MsgX @@(+"Msg"&NN1)
Msg1                          בישראל
Msg2                       בניו-יורק
Msg3                       בוושינגטון
Msg4                                    (c) 1990 רן אברהמי
Msg5         נסה לכתוב בעצמך את המקרו הזה
```

שלב הפעילות הבא שלנו יהיה כתיבת תפריטי פקודות.

<u>תרגיל:</u> בחן את שלוש פיסקאות המקרו 41 \k \0. מהי התפוקה של כל אחת מהן, מה המשותף לכולן ומה שונה בכל אחת מהן.

החשיבה במהלך חיפוש התשובות לתרגיל זה תתרום להבנת הנושאים אשר נדון בהם בהמשך.

## 5.4 שילוב { ... menucall} עם /rl

### .1 <u>Data1</u>

במפת ה-Intra-Macro ישנו איזור לפעילות גלויה. איזור זה מחולק לשני משטחים עיקריים: משטח לכותרות והודעות ומשטח לקליטת נתונים, עדכונים ועיבודים גלויים. בין שני המשטחים הללו יש הפרדה כפולה.

.1 שורת רווח

לשורת הרווח המפרידה בין משטח הכותרות לבין משטח הקלט שתי מטרות: ויזואלית ואופרטיבית. כדי להקל על המשתמש בעת גלגול המסך כאשר משטח הכותרות "מוקפא" מעליו, יש צורך בשורת רווח. אם לא יעשה כן, "מתגלגלות" שורות הנתונים בצמוד לשורת הכותרת התחתונה והמשתמש מאבד מנוחות קריאת החומר.

שורה זו משמשת גם לכתיבת שמות שדות להפעלת משטח הקלט במערך פונקציות או פקודות תפריט של בסיס הנתונים. אחרי כתיבת שמות השדות בעמודות שבשורה זו, למשל ...,x1, x2 יוסתרו השמות על ידי /rfh.

השורה הראשונה של משטח הקלט נמצאת מיד מתחת לשורת הרווח.
את משטח הקלט כולו נגדיר כך: ~rncData1/.

2. הקפאת משטח הכותרות

בין שני המשטחים הללו, כותרות וקלט, נבצע הפרדה על ידי wt/. הקפאה זו מתבצעת כאשר הסמן מוצב בפינה השמאלית העליונה של Data1 בעזרת משפט המקרו ...goto}Data1~/wt}. אם משטח הקלט כולל את עמודה A נבצע את הפקודה wth/. אחרת - נבצע wtb/.

בפרק זה נתייחס למשטח הקלט כולו כאל יחידה הומוגנית אחת שנועדה לקליטת נתונים בכל שורה ובכל עמודה. בפרקים הבאים נראה כיצד מחלקים משטח זה לביצוע מטלות שונות המשלבות קליטת נתונים, עיבודים גלויים ועיבודים סמויים.

משטח קלט שנועד לקליטת נתונים בכל שורה ועמודה מגדיר למעשה את תחום הנתונים בבסיס נתונים (data base) אשר בו כותרות העמודות בלוח הם מגדירי השדות וכל שורה היא רשומה. הצורה ה"ממסדית" שראיתי ברוב היישומים שבחנתי מבוססת על מבנה לוח סגור.

**מבנה לוח סגור**

| פריט | | עלות בש"ח | | | סך הכל | תאריך עדכון |
|---|---|---|---|---|---|---|
| מספר | תיאור | מנהלה | ייצור | שיווק | | |
| 1 | א | 25 | 75 | 100 | 200 | 10.11.90 |
| 2 | ב | 1025 | 575 | 2300 | 3900 | 1. 3.90 |
| 3 | ג | 88 | 112 | 50 | 250 | 15.12.89 |
| 4 | ד | 1500 | 25000 | 500 | 27000 | 9.10.90 |
| סך הכל | | 2638 | 25762 | 2950 | 31350 | |

מבנה זה מחייב הערכה מראש של אורך הלוח, מחזיק עמודות מיוחדות עבור הקווים האנכיים, תופס מקום מיותר בזיכרון לשורות ריקות, מקשה על הוספת וביטול שורות ומקשה על מיונים ועל גישה לנתונים.

המגבלות הללו לא קיימות במבנה לוח פתוח.

**מבנה לוח פתוח**

| מספר | תיאור | מנהלה | ייצור | שיווק | סך הכל | תאריך עדכון |
|---|---|---|---|---|---|---|
| פריט | | עלות בש"ח | | | | |
| סך הכל | | 2638 | 25762 | 2950 | 31350 | |
| 1 | א | 25 | 75 | 100 | 200 | 10.11.90 |
| 2 | ב | 1025 | 575 | 2300 | 3900 | 1. 3.90 |
| 3 | ג | 88 | 112 | 50 | 250 | 15.12.89 |
| 4 | ד | 1500 | 25000 | 500 | 27000 | 9.10.90 |

במבנה הפתוח, שורת "סך הכל" נכללת במשטח הכותרות ובה פונקציות מהסוג @sum. בפונקציות אלו מגדירים את הארגומנט **עד תחתית הגיליון פחות מספר שורות.** הגדרה זו איננה תופסת מקום בזיכרון יותר מאשר ארגומנט על פני שורה אחת. כתוצאה מכך מתקבלים סיכומים אוטומטיים בעת קליטת הנתונים. סיכומים אלה נראים תמיד לעיני המשתמש מכיון שהם כלולים במשטח הכותרת המוקפא על ידי wt/.

המשטח **Data1** מגדיר בדיוק את מטריצת הנתונים על כל עמודותיה. פריט מספר 1 יהיה בשורה הראשונה של משטח זה ופריט מספר 4 יהיה בשורה האחרונה של המשטח. עכשיו נותר רק לקבוע נוהל להגדרה דינמית של Data1 כדי לענות על מצבים של הוספת רשומות (שורות). לשם כך יוכן השטח מראש ויכתב משפט מקרו אחד.

בעת הקמת ה-Intra-Macro מגדירים ~rncData1/ את כל העמודות בשורה אחת שנועדה מראש להיות השורה הראשונה של מטריצת הנתונים.

כדי להגדיר את התחום **Data1** תוך כדי ביצוע המשפט **1_2** נכתוב את משפט השירות:

```
1_2 ...{home}{goto}Data1~/rncData1~.{31}~/ruData1~/wth{return}
31 {end}{d 2}{end}{u}{return}
```

## .2 /ri

הפקודה ri/ מאפשרת הקלדת נתונים בתחום מוגדר, למשל **Data1**. עם הפעלת הפקודה מוגבלת תנועת הסמן ואפשר להציבו רק בתחום שהוגדר (בהמשך אתייחס ל-**Data1**). גם בתוך תחום זה מוגבלת תנועת הסמן לתאים שקיבלו ru/. פקודה זו מאפשרת להקליד נתונים תחת **מקרו פעיל** (CMD Mode). המקרו הבסיסי הוא

**/riData1~**

כדי לצאת מהמצב הזה, מבצעים {esc}.

הבה נפתח את משפט המקרו הזה עד שיהיה פיסקת מקרו שתייצר מגוון תפוקות להקלדת נתונים תחת מקרו פעיל.

כדי שהיישום יהיה נוח להפעלה ונעים לעין, נקדים בכל מקום שאפשר את המלים

{windowsoff}{paneloff clear}{frameoff}

המלה {windowsoff} מקפיאה את המסך. בעת הקלדת הנתונים צריך לאפשר תנועה ידנית של הסמן ולכן יש לכתוב

{windowson}/riData1~

המלים במשפט {paneloff clear} מבצעות שתי מטלות:

1. paneloff - הסתרה של רשימת פקודות התפריט באיזור הדו–שיח (מעל לסרגל העליון) בעת הפעלת המקרו.
2. clear - ארגומנט זה מסתיר את כל mode indicators. כתוצאה, אפשר אמנם להקליד נתונים ולקלוט אותם בגיליון, אבל הם לא ייראו בעת ההקלדה מעל לסרגל, אלא רק אחרי ENTER.

לפיכך יש לכתוב,

{windowson}/riData1{panelon}~

אין צורך, וגם לא רצוי, לבקש {frameon}.

## .3 {menucall ... }

המלה menucall מפעילה נוהל תפריט "עשה זאת בעצמך". יציאה מהתפריט על ידי {esc} מחזירה להמשך קריאת המקרו. כלומר, menucall היא משפט שירות לכל ענין ודבר. נוהל הקמת תפריטים יתואר אחר כך.
בשלב זה נוכל לכתוב

```
\k {restart}{paneloff}{windowsoff}{menucall Menu1}{windowson}
   /riData1{panelon}~{branch \k}
```

המלה {restart} מנקה את המחסנית ומונעת nesting.

**הלולאה ב- \k**

1. {menucall Menu1} – באיזור הדו-שיח רואים את תפריט הפקודות שלנו ששמו **Menu1**.

אפשר לבצע את פקודות התפריט על ידי הצבת הסמן על הפקודה והקשת ENTER, או על ידי הקשת האות השמאלית הראשונה של הפקודה.

2. אם מקישים {esc} מקבלים ~{riData1{panelon/ ואפשר להקיש נתונים ולערוך אותם כאילו לא נמצאים במצב CMD.

3. אם מקישים {esc} מקבלים שוב {menucall Menu1}.

הלולאה הזאת "מתעלת" את המשתמש למצבים מוגדרים מראש: הקלדת נתונים במשטח מוגדר ללא אפשרות של "טיולים" על פני הגיליון ומתן פקודות במערך תפריטים להפעלת הגיליון. הדרך היחידה לצאת מהלולאה היא בהקשת Ctrl-Break (גם זאת ניתן לחסום ע"י {breakoff}).

## 5.5 עץ תפריטים

הקורא כבר יודע להפעיל את נוהל {menu ... } ולכן נוכל לעבור עתה לפיתוח "עץ תפריטים" ב-Intra-Macro הזה.

1. Menu1

תפריט זה מכיל את מילות ההפעלה לביצוע המטלות הראשיות של ה-Intra-Macro כפי שהן מופיעות ברשימת התפוקות.

**לוח הקמת Menu1**

| מלת הפעלה | תיאור הפעולה | פיסקה מפעילה |
|---|---|---|
| שלוף_1 | שליפת ספר הטלפונים של איזור | {\b}{menucall Menu2}{branch \k} |
| חפש_2 | חפש ומצא מספר טלפון לפי שם משפחה ופרטי | {branch 3} |
| מיין_3 | מיין את הנתונים לפי סדר הא"ב | {branch 4} |
| שורה_4 | פתח שורה נוספת לקליטת שם נוסף | {branch 5} |
| הדפס_5 | הדפס את הנתונים שנשלפו | {branch 6} |
| מסך_6 | מסך חדש ללא כל פעולה נוספת | {\b}{branch \k} |
| צא_7 | יציאה לתוכן העניינים ללא כל פעולה נוספת | {branch \m} |
| שמור_8 | שמירת ספר הטלפונים של איזור | {menucall Menu3}{branch \k} |

את מלות ההפעלה, כפי שהן כתובות בעמודה הראשונה בלוח ההקמה שלעיל, יש לכתוב בשורה הראשונה של **Menu1**. בעת הפעלת {menucall **Menu1**} יעלו המלים האלו אל שורת הדו-שיח האמצעית. הצבת הסמן על מלת ההפעלה תעלה את תיאור הפעולה (שייכתב בשורה השניה של **Menu1**) אל שורת הדו-שיח התחתונה. הקשת ENTER תגרום להפעלת הפיסקה המבוקשת. אפשר להפעיל פקודת תפריט גם על ידי הקשת סמל הפקודה.

למיקום מלות ההפעלה בתפריט יש חשיבות. מאחר שברירת המחדל של לוטוס היא "סמן על הפקודה הראשונה משמאל", נכתוב שם את הפקודה השכיחה ביותר. את הפקודה השניה בשכיחותה נכתוב בקצה הימני של התפריט מכיון שדי בהקשה אחת (חץ שמאלה) כדי להציב שם את הסמן. את שאר הפקודות נסדר בשורה זו לפי סדר שנראה לנו הגיוני בעת הקמת היישום.

מלוח הקמה זה נראה בבירור שיש צורך בהסתעפויות לתפריטים נוספים. בעזרת **Menu2** נענה על השאלה המתבקשת "איזה איזור לשלוף" כשמבקשים שליפת נתונים. **Menu3** [K 1]יענה על האפשרויות בעת שמירת נתונים:

א. בטל - לא לשמור.
ב. שמירה בפעם הראשונה.
ג. שמור.

נוכל עתה לערוך את **Menu2**.

**לוח הקמת Menu2**

| מלת הפעלה | תיאור הפעולה | פיסקה מפעילה |
|---|---|---|
| ישראל | ישלפו מספרי טלפון בישראל | {let NN1,"1"}~{branch 1} |
| ניו-יורק | ישלפו מספרי טלפון בניו-יורק | {let NN1,"2"}~{branch 1} |
| וושינגטון | ישלפו מספרי טלפון בוושינגטון | {let NN1,"3"}~{branch 1} |

אם יש יותר משמונה איזורים, הפקודה ה-8 בתפריט תהיה "אחר" וממנה תהיה התפצלות לתפריט נוסף.

## 5.6 תקשורת דיסק - גיליון

פיסקת המקרו ששולפת את הנתונים היא **אותה פיסקה לכל אחד ואחד מהאיזורים.** אלא שלפני שדורשים את הפעלתה "שותלים" את סמל האיזור בתא **NN1** וכתוצאה מכך מקבלים **FileName1** בהתאמה. הבה נראה את השתלשלות העניינים:

### 1. מיפוי Meta-Macro

בעת המיפוי הוחלט ששם קובץ ה-ASCII עם מטריצת הנתונים לאיזור **N** יהיה:

[כל מספר = N] TelN.Tel

הוחלט גם שנסמן,

N = 1 ——> ישראל
2 ——> ניו-יורק
3 ——> וושינגטון

### 2. משפט שירות לשם הקובץ

משפט השירות **FileName1** כתוב במבנה מודולרי. כאשר **NN1** מקבל ערך כל שהוא (למשל "3"), יופיע הערך הזה אוטומטית בשם הקובץ.

```
   Drive1 c:
     Dir1 \Ran\WS_3\WorkShp3
  SubDir1 \Tel
    File1 \Tel
      NN1  <———— | 1 = ישראל    2 = ניו-יורק    3 = וושינגטון |
     Ext1 .Tel
FileName1 +Drive1&Dir1&SubDir1&File1&NN1&Ext1&"{return}"
```

### 3. משפט שירות להודעות

הפונקציה @@ מחזירה את תוכן התא שכתובתו רשומה בארגומנט שלה. נשמע מסובך, אבל הדבר פשוט ואלגנטי והנה ההסבר:

בכתובת **Msg1** כתוב "בישראל", בכתובת **Msg2** כתוב "בניו-יורק" ובכתובת **Msg3** - "בוושינגטון".

```
MsgX @@(+"Msg"&NN1)
Msg1                   בישראל
Msg2                   בניו-יורק
Msg3                   בוושינגטון
```

הארגומנט NN1&"Msg" בתוך הפונקציה @@ מחזיר Msg3 כאשר הערך בתא NN1 (0 ')שווה 3, וכך הלאה. כתוצאה, תחזיר הפונקציה @@ הנמצאת בכתובת MsgX את המלה "בוושינגטון". ואם נכתוב 1 בתא NN1 נקבל בכתובת MsgX את המלה "ישראל" (יש לזכור תמיד שהערכים ב-NN1 חייבים להיות מיושרים לשמאל - left justified labels).

## 4. תפריטי הפקודות

כשמבקשים "שלוף" מקבלים רשימת איזורים.

כשמבקשים "וושינגטון", מתרחשת הפעולה הבאה:

NN1 מקבל את הערך "3" ע"י ~{let NN1, "3"}

FileName1 מחזיר: c:\Ran\WS_3\WorkShp3\Tel\Tel3.Tel

MsgX מחזיר: בוושינגטון

ואחר כך {branch 1}

## 5. המשפט המפעיל

עתה נציג ונסביר את המשפט המפעיל:

```
1 {91}{61}{let XX11,MsgX}~/wtc{goto}YY1~{d}/fitX{esc}{FileName1}~
  {1_2}~{windowson}{branch 41}
```

ואלה הם משפטי השירות שלו:

```
 1_2 /rncParse1~{bs}{u}.{31}~/dpiPares1~oData1~g/re{31}~

  31 {end}{d 2}{end}{u}{return}
  41 {restart}{paneloff}{windowson}/riData1{panelon}~{branch \k}
  61 {paneloff}{windowsoff}{return}
  91 {onerror b_1,XX11}{return}
 b_1 {61}/wtc{home}{goto}Data1~/wth{panelon}{windowson}{Beep1}
     {branch 41}
Beep1 {beep 2}{beep 1}{beep 2}{beep 1}{return}
```

המשפט המפעיל מציב את הערך "וושינגטון" (אם בקשנו ערך זה) בכתובת XX11. כתובת זו הינה המקום המיועד באיזור הכותרות כדי להציג לעינינו הודעות שונות. המשפט ממשיך ושולף את הנתונים לאיזור "וושינגטון" ומעביר אותם לאיזור הפעילות הגלויה. מכאן הוא מעביר את הגיליון למצב הקלדה בפיקוח מקרו עם הלולאה:

/ri <—> {menucall ...}

**תרגיל:** השווה בין Intra-Macro זה לבין המקביל לו בפרק 2.

## 5.7 מעצורים

לתפעול האוטומטי מתלווים סיכונים. האוטומציה, מטבעה, מכניסה את המפעיל להלך רוח "אוטומטי" ואין הוא שם לב לעתים למתרחש. כוחה של האוטומציה כה רב, עד כי בלחיצת כפתור אפשר לגרום נזק בלתי הפיך. לפיכך, חובה על המתכנן להכין "מעצורים" רגע לפני שנזק כזה עשוי לקרות. למשל, בעת שמירת הנתונים - פעולה המבטלת את הקיים וכותבת במקומו את החדש. כאן רצוי להכניס "מעצור" כזה. השימוש בהסתעפויות תפריטים הוא רק אחד האמצעים שלוטוס מעמידה לרשות המשתמש לצורך זה, ולדוגמא:

Menu3 מסתעף מ-Menu1 כאשר מבקשים "שמור".

**לוח הקמת Menu3**

| מלת הפעלה | תיאור הפעולה | פיסקה מפעילה |
|---|---|---|
| לבטל | הפעולה לא תתבצע | {branch \k} |
| חדש | התקנה ראשונה באיזור | {menucall Menu4}{menucall Menu3}{branch \k} |
| שמור | יישמרו הנתונים החדשים במקום הקודמים | {branch 2} |

**תרגיל:** בתרגיל זה שלושה חלקים:

1. התקנת נתונים באיזור בפעם הראשונה **איננה** מחייבת בדוגמא זו פיסקת מקרו נפרדת. הסבר זאת.

2. מלוח הקמת Menu3 רואים שלהתקנה "חדש" יש הסתעפות לתפריט נוסף. ברירת המחדל של תפריט זה צריכה להיות "לבטל". הסבר זאת.
3. מה הם שאר הפריטים בתפריט זה? כיצד ייראה לוח הקמת Menu4 ?

**תרגיל:** שליפת נתונים מאיזור מחייבת הסתעפות לרשימת איזורים. כך גם שמירתם בפעם הראשונה. האם גם שמירתם **שלא** בפעם הראשונה מחייבת הסתעפות כזאת? הסבר.

# פרק 4

# ספריית שירות מסוג mlb

## 1. ספריית שירות

מעגל האוטומציה נסגר כשמשלבים ספריית שירות מסוג mlb בכתיבת Intra-Macro. כשם שמשפט מקרו אחד עשוי לשרת מספר משפטי מקרו אחרים בקובץ wk1 מסוים כך הוא יכול לשרת מספר משפטי מקרו שנמצאים **בקובצי wk1 שונים**. אפשר לשמור קובץ wk1 עם מספר משפטי שירות אוניברסליים ולייבא אותם בפקודה fcc/ לפי הצורך אל הקובץ הנוכחי בעת כתיבת Intra-Macro. אבל אין כתיבה חוזרת של אותו משפט דומה כלל וכלל להפעלתו במבנה מודולרי.

קיימת אפשרות לכתוב משפט מקרו אוניברסלי. נוכל לכתוב, למשל

```
31  {end}{d 2}{end}{u}{return}
```

משפט זה נשמר בקובץ מסוג מיוחד, עם הסיומת mlb (קיצור של Macro Library). את הקובץ הזה ניתן לטעון לזיכרון ללא תלות בטעינתו או אי טעינתו של קובץ wk1 כל שהוא. כלומר, נוצר מצב שבו ניתן לשלוף או למחוק קובץ wk1 ובאותה עת הקובץ, או קבצים, מהסוג mlb ישארו טעונים בזיכרון.

גיליון העבודה wk1 יוצר תקשורת אוטומטית עם כל קובצי mlb שנמצאים במקביל אליו בזיכרון. במגבלות מסוימות מופעלים קובצי mlb הללו כחלק אינטגרלי לכל דבר של הגיליון הנוכחי. משהוחלף הגיליון באחר, נשאר קובץ ה-mlb בזיכרון ונותן לו שירות בדיוק כשם שנתן לקודמו. כשמקפידים על כתיבה מודולרית יהיו קובצי ה-mlb קטַנים יחסית, ולא יחזיקו מקום רב בזיכרון.

את השירות הניתן על ידי קובץ mlb מסוים אפשר לסווג לפי **רמת השירות:**

1. שירות לכל קובצי wk1.
2. שירות לקובץ wk1 שחוזר ו"עולה" למסך לעתים מזומנות כמו, למשל, קובץ תוכן העניינים של היישום.
3. שירות למספר קובצי wk1.
4. שירות לקובץ wk1 בודד.

## 2. שירות על ידי קובץ mlb

את פיסקאות המקרו האוניברסליות שומרים בקובץ ServAll.Mlb, אשר טוענים אותו לזיכרון באופן אוטומטי עם הפעלת היישום. הנה חלק מהתכולה של קובץ זה:

```
      1  {99}{restart}{71}{menucall Menu1}{windowson}/riData1{panelon}~{branch 1}
     31  {end}{d 2}{end}{u}{return}
     32  .{end}{d 2}{end}{u}{return}
     35   X{esc}{return}
     41  {99}{restart}{71}{windowson}/riData1{panelon}~{branch 1}
     51  {99}/frMenu~{quit}
     61  {indicate Msg1}{return}
     62  {indicate Msg2}{return}
     63  {indicate Msg3}{return}
     64  {indicate Msg4}{return}
     69  {indicate Msg901}{return}
     71  {frameoff}{paneloff clear}{windowsoff}{indicate Msg901}{return}
     72  {panelon}{windowson}{return}
     81  {blank XX11}~{return}
     91  {onerror 91_1,XX11}{return}
   91_1  {71}/wtc{home}{goto}Data1~/wtb{72}{Beep2}{branch 41}
     99  {return}
  Beep1  {beep}{beep 2}{return}
  Beep2  {beep}{beep 2}{beep}{beep 2}{return}
   Msg1                                  ה מ ת ן
   Msg2           ל א  נ ק ל ט . . .  ת ק ן  א ת  ה ש ג ו י ם
   Msg3      ת ק ל ה  ב מ ד פ ס ת . . .  ת ק ן  ו ה ק ש  E N T E R
   Msg4
 Msg901                                                     (C) רן אברהמי
```

משפטי מקרו אלה מוכרים מהפרקים הקודמים. מקרו **99** "מחזיק מקום" (place holder) למשפט שאנו עשויים לרצות לכתוב בתחילתו של כל מקרו, כמו למשל {breakoff}.

לשמירת המקרו של הקובץ Menu.Wk1 בקובץ Menu.Mlb נבחן שוב את "המקרו השולף" מפרק 1.

```
 \z \0 {restart}{windowson}{paneloff clear}{frameoff}{indicate Msg901}
       {home}{?}{windowsoff}/c{bs}{r 2}~xData1~{onerror \z}
       {if @cell("type",xData1)="b"}{branch \z}
       fr
xData1
       ~{branch \z}
    \a {indicate}/wtc/wgpd/rfrHide1~{quit}
    \b {indicate}/rfhHide1~/wgpe/wtc{home}{goto}XX1~/wtb{quit}
```

<u>הסבר:</u>

1. כותבים מקרו ‎\z \0‎ חדש שנשאר בגיליון wk1 שמכיל את מסך תוכן העניינים:

```
\z \0 {branch mz}
```

2. משנים את שמות שאר פיסקאות המקרו כדלקמן,

```
\z  \0 ——> mz          \a ——> ma          \b ——> mb
```

3. אי אפשר לבצע פקודת העתקה ‎c/‎ בין גיליון wk1 לבין קובץ mlb, אבל אפשר לבצע העברת נתונים בין שני סוגי הקבצים הללו באמצעות פקודות מקרו. בתוכן העניינים ישנו המשפט הבא:

```
/c{bs}{r 2}~xData1~
```

תפוקתו היא העתקת תוכן התא הנמצא שתי עמודות ימינה ממיקומו של הסמן ובאותה שורה, אל כתובת xData1. פעולה זו תגרור הודעת שגיאה (ERROR), אם xData1 נמצא בקובץ mlb. לפיכך עלינו לפעול כך:

א. נוותר על המלים ‎fr/‎ בפיסקת המקרו המקורית.

ב. נחליף את המשפט

```
/c{bs}{r 2}~xData1~
```

במשפט הבא:

```
{let xData1,+"/fr"&@@(+"d"&@string(@cellpointer("row"),0))}~
```

נפרק את המשפט הזה למרכיביו ונבחן אותם זה אחר זה.

הפונקציה ‎@cellpointer("row")‎ מחזירה את מספר השורה, למשל - 8.

הפונקציה ‎@string(@cellpointer("row"),0)‎ מחזירה את מספר השורה כתווית (label).

תוכן העניינים נמצא בעמודה b ועליה מציבים את הסמן כדי לבחור את הקובץ לשליפה. שתי עמודות ימינה, בעמודה d, כתוב המקרו שמיועד לבצע את הפעולה המבוקשת. לפיכך, השילוב הבא מחזיר את הכתובת המדויקת של התא הנמצא שתי עמודות ימינה מהמקום שבו נמצא הסמן:

```
+"d"&@string(@cellpointer("row"),0)          ——> d8
```

התוספת הבאה מחזירה את תוכן התא. למשל, אם בכתובת d8 כתוב Ran5 אזי נקבל:

```
@@(+"d"&@string(@cellpointer("row"),0))      ——> Ran5
```

עכשיו נותר רק להוסיף fr/ כדי לקבל

+"/fr"&@@(+"d"&@string(@cellpointer("row"),0)) ——> /frRan5

התוצר הזה של הארגומנט השני במקרו { ... let} ייכתב בכתובת **xData1** הנמצאת בקובץ mlb, ולא בגיליון עצמו.

4. את המקרו הבא נשמור בקובץ mlb (בספריית מקרו), כולל שמות תחום.

<u>הערה:</u> את כל ארבע השורות הראשונות של **mz** רצוי לכתוב **בתא אחד.**

```
    mz {restart}{windowson}{paneloff clear}{frameoff}{indicate Msg901}
       {home}{?}{windowsoff}
       {let xData1,+"/fr"&@@(+"d"&@string(@cellpointer("row"),0))}~
       {onerror \mz}{if @cell("type",xData1)="b"}{branch \mz}
xData1
       ~{branch \mz}
    ma {indicate}/wtc/wgpd/rfrHide1~{quit}
    mb {indicate}/rfhHide1~/wgpe/wtc{home}{goto}XX1~/wtb{quit}
```

עכשיו נחזור להסביר את השמות שנתנו לפיסקאות המקרו השונות. אם אותו שם של מקרו כתוב במספר קובצי mlb וכולם נטענו לזיכרון, יש קדימות לשם שבקובץ mlb שנטען ראשון.

נאמר ששם תחום כל שהוא מוגדר גם ב-wk1 וגם ב-mlb ונאמר, שהוא משולב במשפט מקרו, למשל עם {...let}. אזי ל-wk1 יש קדימות וה-let יפעל על הגיליון ולא על ה-mlb.

כדי לאפשר שימוש נרחב בספריות מקרו יש להקפיד על ייחודיות שמות המקרו בקבצים השונים. אני נוהג לתת לקובצי ה-wk1 שמות שהם מספרים ולמקרו שבתוכם - שמות שהם מספרים. כשמכינים מערך להפעלה בספריות מקרו, נותנים למקרו שמות משולבים. למשל, המקרו ששמו **1_1** נמצא בקובץ Ran1.wk1, ואילו המקרו **3_1** נמצא בקובץ Ran3.wk1.

פרט לקובץ תוכן העניינים, כאן הקידומת היא **m** (בשביל Menu) ולפיכך z\ הופך mz וכך הלאה.

קיימת אפשרות של הפעלה משולבת עם מספר קובצי mlb שנטענו לזיכרון. בדוגמאות שלפנינו נטענו לזיכרון שני קבצים:

1. ServAll.mlb המכיל את ספריית המקרו האוניברסלית.
2. Menu.mlb המכיל את ספריית המקרו עבור קובץ תוכן העניינים Menu.wk1.

בקובץ Menu.mlb ישנו המקרו {indicate Msg901}. **Msg901** עצמו נמצא ב-ServAll.mlb, וכתוצאה מכך נותן הקובץ servAll.mlb שירות לקובץ Menu.mlb, שנותן שירות לקובץ Menu.wk1.

למעשה, כאשר טוענים מספר קובצי mlb לזיכרון, הם פועלים במשולב **כאילו היו כולם קובץ mlb אחד.**

## 3. טעינה אוטומטית

### 3.1 טעינה אוטומטית של שירות אוניברסלי

יישומי לוטוס שונים מוקמים בנוהלים שונים ועל כן ייתכן שיהיו יישומים בנוהל mlb ויישומים שלא בנוהל זה. כדי לשמור על עיקרון האוטומציה בכניסה אל תוכן העניינים של סביבת לוטוס תחת mlb, צריך לבצע טעינה אוטומטית שני קובצי השירות האוניברסליים - ServAll.mlb ו-Menu.mlb, לפני ששולפים את תוכן העניינים Menu.wk1 אל המסך.

לשם כך נקים קובץ בשם xMenu.wk1, אשר יישלף **לפני Menu.wk1** ואשר יכיל רק את המקרו הבא:

**{FileName1}**~l**{FileName2}**~q{branch 51}

```
       \0 {if @isapp("MacroMgr")}/acaMacroMgr~1iMacroMgr~1
          /aaMacroMgr~1iMacroMgr~l{FileName1}~l{FileName2}~q{branch 51}
   Drive1
     Dir1
     File1 ServAll{return}
   Drive2
     Dir2
     File2 Menu{return}
FileName1 +"X{esc}"&Drive1&Dir1&File1
FileName2 +"X{esc}"&Drive2&Dir2&File2
```

כדי לאפשר המשך עבודה בצורה נכונה יש לבדוק אם נטענו לזיכרון ספריות מקרו כלשהן ולהסירן. אחר כך יש לטעון את שני הקבצים שישרתו בהמשך העבודה את כל מרכיבי היישום שיוקם.

במהלך טעינת ה-MacroMgr מצמידים אחד ממקשי F כדי לאפשר אחר כך את הפעלתו האוטומטית דרך מקרו. במשפט המקרו שלמעלה אנחנו מצמידים ל-MacroMgr את המקש F10. כתוצאה מכך אפשר יהיה במשפטי מקרו אחרים להפעיל את ה-MacroMgr ע"י המלה {app4}.

**תרגיל:** 1. כתוב משפט מקרו לטעינה אוטומטית של קובץ mlb המשרת קובץ wk1 מסוים.

2. כתוב פיסקת מקרו שתבטיח הפעלת כל קובצי ה-wk1 ביישום הנוכחי באמצעות המקרו 0\. מדוע הכרחי לכלול מקרו זה בקובץ xMenu.wk1? איזו פיסקה תופעל ראשונה: טעינת MacroMgr, או ביקורת ההפעלה תחת 0\?

## 3.2 טעינה ופריקה אוטומטית של שירות מיוחד

ספריית מקרו לגיליון wk1 אחד בלבד, או לקבוצה של גיליונות (אך לא לכל הגיליונות בסביבת היישום הנוכחית) היא בחזקת **שירות מיוחד.** יש לטעון אותה אל הזיכרון כל אימת שידרשו שירותיה ויש **לפרוק אותה מהזיכרון כשאין בה צורך.** פעולות אלו מתבצעות על ידי פיסקאות המקרו הבאות.

הטעינה האוטומטית מתרחשת אחרי שליפת הגיליון (נאמר Ran5.wk1) באמצעות המקרו 0\. בדוגמא זו נטען קובץ Ran5.mlb ומכיון שהוא נמצא בדיסק, צריך לקרוא בשמו המלא.

לפריקה אוטומטית **מהזיכרון** די בשם הקובץ ללא ציון הנתיב השלם.

```
     \z \0 {app4}lX{esc}{FileName1}~q{branch 41}
        \m {app4}r{FileName2}~{branch 51}
    Drive1
      Dir1
     File1 Ran5{return}
 FileName1 +Drive1&Dir1&File1
 FileName2 +File1
```

## 4. יתרונות

שימוש בספריית מקרו במבנה mlb כולל יתרונות חשובים. בראש ובראשונה מורחב כאן עיקרון המודולריות אל מימד מרחבי חשוב נוסף.

מקרו משותפים למספר קובצי wk1 כותבים פעם אחת בלבד ושומרים בספריית שירות אוניברסלית. עושים זאת על פי העיקרון שמנחה אותנו לכתוב בגיליון משפט אחד בלבד לשירותם של משפטים רבים אחרים באותו גיליון. בנוסף לכך, מעניק נוהל זה, בשילוב עם נוהל שיוסבר בפרק האחרון, הגנה מפני פגיעה מקרית של משתמש במקרו, מפני "מציצנים" וגם מפני פירטים.

# פרק 5

# השליטה בדיסק

## 1. סוגי נתונים בשליטת לוטוס

תוכנת לוטוס יכולה לנהל שלושה סוגי קבצים: mlb, ASCII ו-wk1. קובצי הגיליון (wk1) וקובצי ספריית המקרו (mlb) הם חלק ממערך לוטוס וניתן לייצרם ולהפעילם באמצעותו בלבד. התוכנה גם מאפשרת תקשורת בין קבצים אלה.

קיימים נוהלים בלוטוס לעדכון הקובץ הנוכחי מנתונים הנמצאים בקובץ שנמצא בדיסק. תא מסוים על המסך "קורא" את הנתון מתא מסוים בקובץ wk1 אחר על הדיסק. אפשר גם לעדכן את הקובץ שנמצא על המסך מקובץ wk1 אחר על הדיסק בנוהל fc/. עדכון בכיוון ההפוך איננו אפשרי. כלומר, אי
אפשר "לשלוח" נתון מתא כלשהו בקובץ wk1 שעל המסך אל תא מסוים בקובץ wk1 אחר על הדיסק.

קיים "עולם" נוסף על הדיסק - עולם קובצי **ASCII**. כל יישום מקצועי מסוגל לייבא ולייצא קובץ ASCII "נקי" (ללא תווי ההוראות המיוחדים ליישום). יישומים אלה מקיפים מגוון של מטלות - עיבוד תמלילים, הנהלת חשבונות, ניהול מאגרי נתונים, גיליונות חישוב, תקשורת מחשבים ועוד כהנה וכהנה.

ללוטוס נוהלים פשוטים ונוחים המקנים למשתמש שליטה כמעט בלתי מוגבלת בקובצי ASCII. שליטת הלוטוס בדיסק, בקבצים שנוצרו בלוטוס ובקבצים שנוצרו על ידי תוכנות אחרות, באה לידי ביטוי מלא כאשר משלבים נוהלי לוטוס לשליטה בקובצי ASCII עם **Meta-Macro** ועם **Intra-Macro**. לא פעם הפעלתי לוטוס כתוכנת-על לגישור בין שתי תוכנות אחרות. למשל, יש יישומים לניהול בסיסי נתונים השומרים תאריך במבנה מסוים, השונה מזה של יישומים אחרים. בעזרת לוטוס אפשר לשלוף את התאריך שנוצר על ידי תוכנה **א**, לשנות אותו למבנה תואם לתוכנה **ב** ולשתול את התוצאה במקום המתאים בתוכנה **ב**.

בנוסף לכך, הניסיון מראה ששמירת נתונים במבנה ASCII והפעלתם בגיליון Intra-Macro נותנת חיסכון רב במקום על הדיסק ומשיגה מהירות גבוהה לשמירה, שליפה ועיבוד נתונים.

החשיבה המרחבית, המשוחררת מהחזות של סרגלי הלוטוס, או של מבנה סגור של תוכנה כל שהיא, פותחת אפשרויות רבות. דומה, כאילו **נשלף הדיסק**

**כולו אל בין סרגלי הלוטוס על המסך.** את מירב אפשרויות ההפעלה משיגים כאשר פועלים על פי העקרונות הבאים:

1. שומרים נתונים, מספרים ו/או טקסט, אך ורק במבנה ASCII.

2. עיבודים מבצעים ב-Intra-Macro ושומרים אותם רק במקרים מיוחדים.

3. היישום כולו מופעל על ידי קובצי wk1 קטנים מאוד ומהירים

4. מבצעים מיפוי הדיסק ל-Meta-Macro.

5. מבצעים אוטומציה מירבית של:
   - הכניסה למחשב
   - ניהול הדיסק
   - ניהול מטלות תוך גיליוני ב-Intra-nacro

## 2. מטריצות נתונים ב-ASCII

במרבית היישומים יש לוחות עם נתונים. הלוח מורכב משני חלקים עיקריים: כותרות ונתונים. המשטח המלבני בלוח שמכיל את הנתונים הוא המטריצה שתישמר במבנה ASCII. כותרות הלוח, בדרך כלל, יישמרו בקובץ wk1, כשהן משולבות ב-Intra-Macro.

את משטח הנתונים אפשר לחלק למשטחי משנה בכל צורה שהיא ולשמור כל תת-משטח כזה בקובץ ASCII אחר. במטריצות הללו אפשר לשלוט באופנים הבאים:

פקודות התפריט fi/ pf/. פקודות אלו מחייבות שליטה במטריצות שלמות מכיון שהן מייבאות או מייצאות משטח של נתונים בשלמותו. שיטה זו מוגבלת בגודל הגיליון או בגודל הזיכרון, מי שקודם.

משפטי מקרו מהסוג {open ... }. משפטי מקרו אלה מאפשרים שליטה בדיסק **עד רמת התו הבודד** לפעולות שמירה ולשליפה. שיטה זו מאפשרת ייבוא וייצוא סלקטיבי של תווים, רשומות, או חלקי מטריצות אל הגיליון וממנו. כתוצאה מכך נוצרת יכולת שליטה בקובצי ASCII **בגודל אין סופי**, משוחררת ממגבלות גודל הגיליון והזיכרון.

השליטה במטריצות הנתונים מתבצעת בשני מישורים: תקשורת עם הדיסק ועיבודים בגיליון. לשם כך מעמיד לוטוס לרשות המשתמש את הכלים הבאים:

1. פקודת התפריט fin/ לשליפת הנתונים שנשמרו ב-ASCII אל תוך הגיליון במבנה של value, המציבה כל שדה מספרי מוצב בעמודה נפרדת.

2. שילוב בין fit/ לבין dp/ לפיצול אוטומטי של שדות טקסט ו/או מספרים לעמודות בגיליון.

3. סידרה של פונקציות @ לטיפול במחרוזות טקסט.

נתונים נוצרים בצורות שונות וממקורות שונים. המבנה המשמש להצגתם - הלוח - זהה בכולם. הבה נבחן מערך נתונים לדוגמא.

**לוח מספר 1**

**תקציבי הרשויות לפי סעיף תקציבי - תכנון לעומת ביצוע**

± % : תכנון פחות ביצוע ב-% מתכנון ... - חריגה + עודף

* חריגה מתקציב

תקציבי הרשויות באלפי ש"ח

| הרשות | | סה"כ | | | סעיף א | | | סעיף ב | | |
|---|---|---|---|---|---|---|---|---|---|---|
| שם | סוג | תכנון | ביצוע | ± % | תכנון | ביצוע | ± % | תכנון | ביצוע | ± % |
| * בני גד | ממ | 771 | 873 | -13.2 | 578 | 623 | -7.8 | 193 | 250 | -29.5 |
| גנים | מא | 1645 | 1544 | 6.1 | 1234 | 1123 | 9.0 | 411 | 421 | -2.4 |
| * תמר | ממ | 47 | 55 | -17.0 | 35 | 45 | -28.6 | 12 | 10 | 16.7 |
| * אבן פינה | מא | 7719 | 8962 | -16.1 | 5789 | 6789 | -17.3 | 1930 | 2173 | -12.6 |
| דור | ממ | 1039 | 962 | 7.4 | 779 | 750 | 3.7 | 260 | 212 | 18.5 |

בלוח זה התווים המודגשים הם נתוני קלט, השאר הם עיבודים וחישובים. ברמת הפירוט הגבוהה ביותר יש כאן שישה וקטורים (טורים) עם נתוני קלט לשמירה, לעדכון ולשליפה:

1. שם הרשות
2. סוג הרשות
3. סעיף תקציבי א - תכנון
4. ביצוע
5. סעיף תקציבי ב - תכנון
6. ביצוע

לצורך הדוגמא הנוכחית נאחד את הוקטורים האלה לשלושה "משטחים" (או מטריצות) ונקרא לכל משטח בשם כלשהו:

1. סעיף תקציבי א - תכנון וביצוע = **Print1**

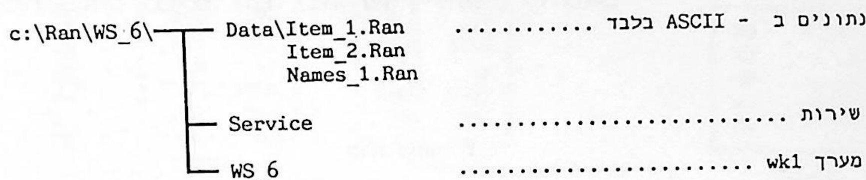
2. סעיף תקציבי ב - תכנון וביצוע = **Print2**

3. פרטי הרשות = **Names1**

עד כאן מבנה תוך גיליוני. עתה נקים את התקשורת עם הדיסק.

מיפוי ל-Meta-Macro

סביבה ליישום לוטוס - סדנא 6




כתובת הדיסק עבור המטריצה **Print1**:

c:\Ran\WS_6\Data**Item_1**.Ran

עבור המטריצה **Print2**:

c:\Ran\WS_6\Data**Item_2**.Ran

עבור משטח הנתונים **Names1**:

c:\Ran\WS_6\Data**Names_1**.Ran

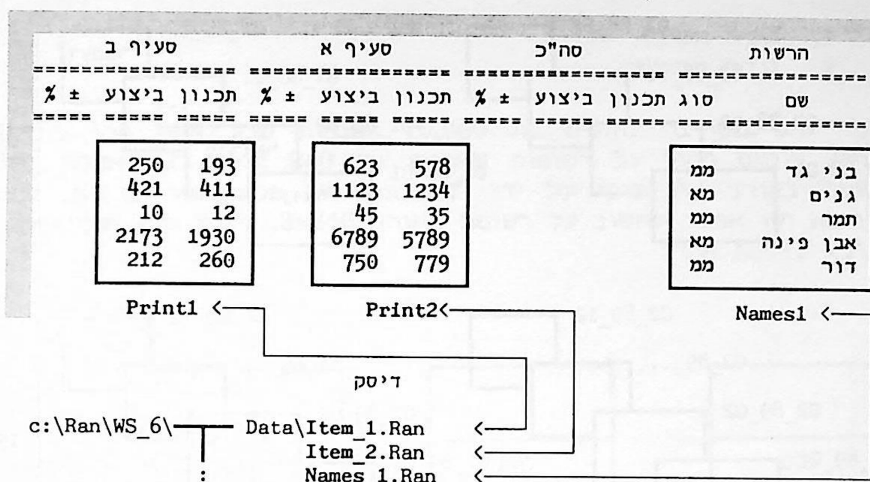
עכשיו נקים מבנה רב מימדי בתוך הדיסק עבור הנתונים שבלוח 1.

נגדיר תחילה וקטורים (מימדים) המאפיינים את הנתונים. לדוגמא:

1. רשות
2. סעיף תקציבי
3. סטטוס
4. חודש בשנה
5. שנה

נאמר, שהמטריצה הקטנה ביותר היא PrintN [כל מספר = N > 0].

**מ ס ך**




אפשר לקבל מטריצה כזאת לכל **סעיף** לכל **חודש** לכל **שנה**.
אם כל מטריצה שמורה בקובץ ASCII, נוכל לתאר לעצמנו שהכתובת של קובץ זה בנויה מספרות, למשל:

01_90_11.Ran

כאשר,

- שתי הספרות הראשונות משמאל מסמלות את הסעיף התקציבי.

- שתי הספרות במרכז השם מסמלות את השנה.

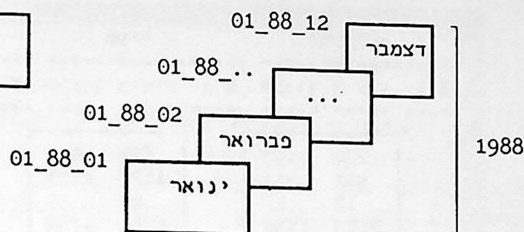
- שתי הספרות בימין השם מסמלות את החודש בשנה.

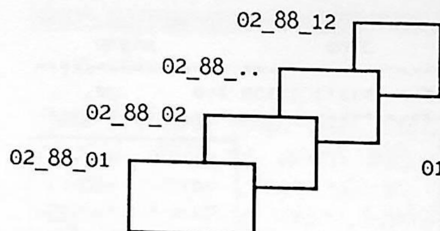
**אין צורך להקים בפועל** את מערך הקבצים. במבנה Intra-Macro קובץ נפתח אוטומטית ברגע שמופיעים הנתונים הראשונים המאופיינים ע"י ההגדרות שלמעלה. וזאת מכיון ששם הקובץ הוא משוואה במבנה של משפט שירות המשרת בו-זמנית את המשפט ה"שומר" ואת המשפט ה"שולף" את הנתונים.

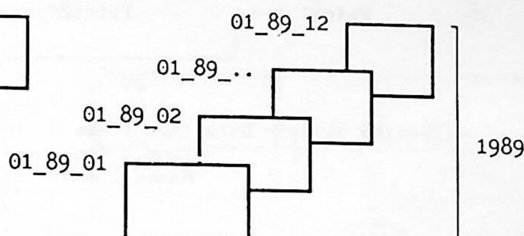
התרשים הבא ממחיש את הדברים. במהלך הפעלה שוטף של המערכת יהיו בתרשים זה "חורים", שמשמעותם מטריצה ריקה (אין נתונים במשבצת זו או קובץ לא קיים עדיין). לנו אין זה מפריע כלל. להיפך, ה"חורים" הללו חוסכים במקום מבלי שהמבנה התפישתי יושפע מכך.

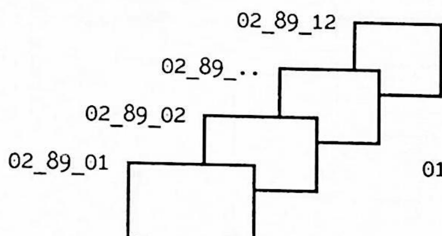
סעיף תקציבי מספר 1

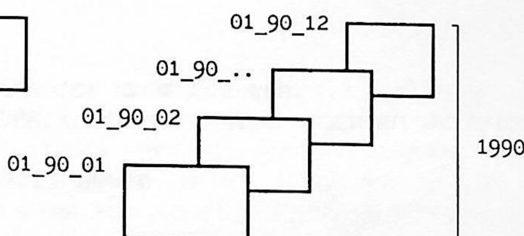
סעיף תקציבי מספר 2

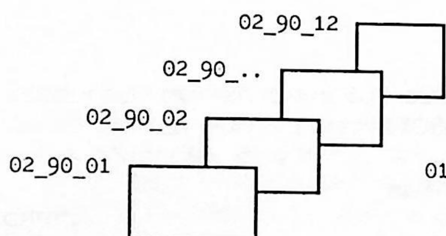



בדרך זו אפשר להגדיר מספר רב של מאפיינים ולהשתמש לשם כך בכל 11 התווים של שם הקובץ. אפשר ולפעמים אף רצוי, לאפיין משתנים ע"י שם המחיצה. למשל, \Data_1\
\Data_2\

הקשר בין הגיליון לבין הדיסק מותנה, אם כך, בקיומם של שני מרכיבים:

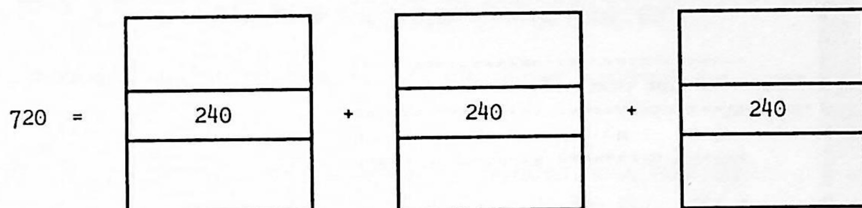
1. מיפוי Meta-Macro.

2. הגדרת מאפייני מטריצת הנתונים במבנה המאפשר פתיחת קבצים (מתן כתובת) אוטומטית ב-Intra-Macro.

השליטה על רשומות בודדות ועל תווים בודדים בקובצי ASCII פותרת בעיות של גודל בשני תחומים:

1. גודל הקובץ
2. גודל הרשומה

נאמר שיש מטריצת נתונים בת **10,000** רשומות שכל אחת מהן בת **720** תווים. לוטוס פועל על רשומות שאורכן עד **240** תווים, אך אפשר להקים רשומה באורך אין סופי על ידי "משיכתה" על פני מספר קבצים. למשל, בדוגמא הזו אפשר לחשוב על רשומה באורך **3x240**, כשכל קטע שלה מאוחסן בקובץ ASCII אחר.




נחזור לבחון את אורך המטריצה. לביצוע עיבודים על המטריצה כולה אפשר לייבא אותה בחלקים. לעיתים קרובות נדרשים עיבודים על תווים בודדים או על רשומות המוגדרות בקבוצת איפיון אחת. לצורך זה "מייבאים" תו אחד, קבוצה של תווים, רשומה אחת, או קבוצה של רשומות, מבצעים את העיבודים ו"שותלים" את התוצאות בכל מקום שרוצים על הדיסק. כך שאפשר לחשוב על גודל קובץ אין סופי.

## 3. הקמת רשומה מנתונים מעורבים

אפשר לראות כל מטריצה כבסיס נתונים המורכב משדות ורשומות. בתכנון מבנה הקובץ והרשומה רצוי לשמור על המבנה הקלסי של שדות ורשומות בבסיס נתונים. אין צורך בתו מפריד (delimiter) בין השדות ברשומה והם יכולים להיות צמודים. אבל חייבים לשמור על מספר תווים שווה בשדה נתון בכל רשומה. לשם כך נבצע הקמת רשומה מנתונים מעורבים.

**הקמת רשומה מנתונים מעורבים**

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | תקציב לחודש ינואר 1990 | | | | | | | |
| 2 | | | | | | | | | | |
| 3 | | ====== | ========= | =========== | ==== | | | | | |
| 4 | | סעיף תקציבי | | מחלקה | | | | | | |
| 5 | | ====== | ========= | =========== | ==== | | | | | |
| 6 | | ב | א | שם | מספר | | | | | |
| 7 | | ====== | ========= | =========== | ==== | | | | | |
| 8 | | | | | | | | | | |
| 9 | | 1,125 | 123,654 | ייצור | 1 | | | | | |
| 10 | | 206 | 6,333 | שיווק | 2 | F10 | G10 | H10 | I10 | J10 |
| 11 | | 225 | 24,731 | מינהלה | 3 | | | | | |

לפנינו לוח עם נתונים מעורבים (טקסט ומספרים) ועם דרישה למספר תווים שונה בכל טור. למספר המחלקה דרושים 4 תווים, לטקסט בעמודה "שם" דרושים 11 תווים, למספרים ב"סעיף תקציבי א" דרושים 9 תווים ולמספרים ב"סעיף תקציבי ב" דרושים 6 תווים. רוצים להקים רשומת ASCII לכל אחת מהשורות בלוח הזה במבנה הבא.

אורך הרשומה = 30 = 4 + 11 + 9 + 6

מבנה הרשומה יהיה:

| | | | | |
|---|---|---|---|---|
| מספר סידורי של השדה | 4 | 3 | 2 | 1 |
| | 2 | שיווק | 6333 | 206 |
| מספר תו ראשון בשדה | 26 | 15 | 6 | 0 |
| גודל השדה (מספר תווים) | 4 | 11 | 9 | 6 |

משוואה פשוטה מייצרת את הרשומה הזאת. נפיק תחילה מחרוזת תווים לשדה מספר 1 (סעיף תקציבי א) עבור רשומה מספר 2 (מחלקת שיווק). הנתון המקורי למשבצת זו נמצא בכתובת **B10** במבנה של **value** עם שלוש הספרות "206". התפוקה המבוקשת היא מחרוזת (**string**) תווים בת שישה תווים: "   206". כדי לקבל מחרוזת זו נכתוב בכתובת **F10** את המשוואה הבאה:

```
@repeat(" ",6-@length(@string(B10,0)))&@string(B10,0)
```

למשוואה זו נקרא **משוואת השדה.**

כדי לקבל את המחרוזת לשדה הזה עבור רשומה מספר **1**, נכתוב בכתובת **F9** את המשוואה הבאה:

```
@repeat(" ",6-@length(@string(B9,0)))&@string(B9,0)
```

כדי לקבל את המחרוזת לשדה הזה עבור רשומה מספר **3**, נכתוב בכתובת **F11** את המשוואה הבאה:

```
@repeat(" ",6-@length(@string(B11,0)))&@string(B11,0)
```

**<u>תרגיל:</u>** העתק את הלוח שבדוגמא אל גיליון לוטוס, כתוב בו את שלושת המשוואות הללו ובחן את תפוקת המשוואות כאשר מבצעים את הדברים הבאים,

1. משנים את הנתונים בלוח.
2. במקום תו הרווח " " בפונקציה **repeat** כותבים "0" או כל תו אחר.
3. במקום המספר 6 בפונקציה **repeat** כותבים מספרים אחרים.
4. במקום המספר 6 בפונקציה **repeat** כותבים כתובת של תא כל שהוא בגיליון. כתוב בתא זה את המספר **6**, את המספר **11**, כל מספר.

כתוב בכתובות המתאימות את משוואות השדה לשלושת השדות האחרים.

נותר לנו עכשיו לחבר את התפוקות של ארבע משוואות השדה הללו לרשומה אחת. כדי לחבר את המחרוזות לרשומה כולה, נכתוב בכתובת **J10** את המשוואה הבאה:

```
+F10&G10&H10&I10
```

למשוואה הזאת נקרא **משוואת הרשומה.**

**תרגיל:** 1. כתוב את משוואת הרשומה לכל שלוש הרשומות בלוח.

2. הגדר חמישה תחומים, אחד לכל אחת מחמשת העמודות עם משוואות המחרוזות וכתוב משפט מפעיל אחד בלבד (עם **/pf**), משפט שירות אחד (עם **{open ...,w}**) ומשפט שירות אחד מסוג **FileName1** כך שנקבל:

אחרי הקלדת מספר (**1**, **2**, **3**, **4**, או **5**) בכתובת כל שהיא בגיליון ואחרי הפעלת המקרו ישמרו הנתונים מהתחום שביקשת, בקובץ שנועד להם בדיסק.

משנתקבלו כל ההחלטות, מן הראוי לרשום אותן בצורה מסודרת. רישום זה יהיה מרכיב מרכזי בתיעוד של היישום ויעזור הרבה בהמשך הפיתוח. את מבנה הרשומה נרשום בטופס מיוחד, כמו בדוגמא שלפנינו:

**טופס מבנה רשומה**

תקציבי מחלקות לחודש ינואר 1990 לפי סעיף תקציבי.

שם הקובץ ... c:\Ran\WS_5\Data1\01_90.Ran

גודל הרשומה = 32 = 2 + 30

| השדה | | | |
|---|---|---|---|
| מספר | תיאור | y) | @mid(x, |
| 1 | סעיף תקציבי: א | 6 | 0 |
| 2 | ב | 9 | 6 |
| 3 | מחלקה: שם | 11 | 15 |
| 4 | מספר | 4 | 26 |

את הטופס הזה מכינים בגיליון לוטוס, כשכל מרכיב נמצא בעמודה משלו ואפשר לשלב בו נוסחאות כדי להקל על הכנתו.

כפי שכבר ציינתי, יש בלוטוס קבוצת פונקציות לטיפול במחרוזות תמליל. @mid(Range,x,y) היא אחת הפונקציות החשובות בקבוצה זו. היא שולפת y תווים מתוך מחרוזת תמליל (רשומת ASCII שנשלפה אל הגיליון, למשל) הנמצאת בכתובת **Range** החל בתו מספר x (כולל תו זה). בטופס מבנה רשומה, y הוא מספר התווים בשדה (גודל השדה) ו-x הוא מספר התו הראשון בשדה. מכאן רואים בנקל, שאם i הוא מספר השדה אזי,

x[i] = x[i-1]+y[i-1]

גודל הרשומה הוא סכום מספר התווים בכל השדות (6+9+11+4=30). לעיתים נוסיף עוד שני תווים לרשומה, שמשמעותם היא סוף רשומה (End of Line) וחזרה (Carriage Return) ומכאן שני התווים הנוספים. הסיבה היא, ששתי פקודות לוטוס מוסיפות את התווים הללו: הפקודה /pf (אפשר לשנות זאת בברירת המחדל של הגיליון) והפקודה {writeln ... }. תוספת שני תווים אלה מאפשרת ליצור קובץ ASCII השומר על חזותה של המטריצה שנכתבה אליו.

# 4. כתיבה אל הדיסק

## 4.1 כתיבת רשומה אחת

כדי להדגים כתיבה של רשומה אחת אל הדיסק, נניח שביצענו שתי פעולות:

- שלפנו לגיליון את נתוני התקציב לחודש ינואר.
- הגדרנו את העמודה המכילה את משוואות הרשומות:

/rncPrint2~j9.j11~

נניח עתה כי רוצים לעדכן על הדיסק נתונים של רשומה אחת בלבד: לשנות במחלקת **השיווק** את הסכום **בסעיף תקציבי מספר 2** לסכום גדול יותר. אנו נשנה את הנתון על ידי הקלדה רגילה בלוח ואחר כך תעודכן בדיסק **הרשומה השניה בלבד** על ידי **\b**.

```
        \b {getnumber Msg1,NN2}~
           {open FileName1,m}{open FileName1,w}
           {setpos (NN2-1)*Length1}
           {writeln @index(Print2,0,NN2-1)}
           {close}
           {quit}
       NN2          <——————— מספר המחלקה
   Length1 32       <———————————————————— FileName1 גודל רשומה ב
 FileName1 +Drive1&Dir1&File1&Ext1
    Drive1 c:\
      Dir1 Ran\WS_5\Data1\
     File1 01_90
      Ext1 .Ran{return}
      Msg1         (1..3) הקש את מספר המחלקה -->
```

**<u>הסבר:</u>**

במשפט הראשון מבקשים מהמשתמש להקיש את מספר המחלקה שברצונו לעדכן על הדיסק. מספר זה נקלט בכתובת **NN2**.

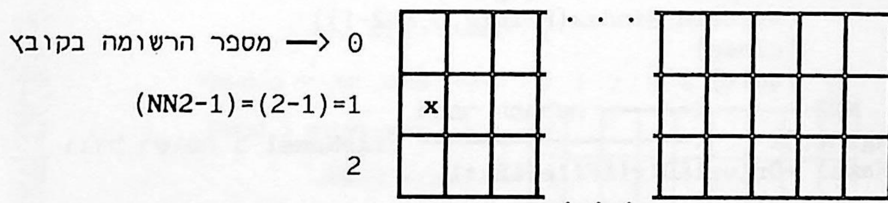
לאחר מכן יש שורה עם שני משפטים:

המשפט {open **FileName1**,m} פותח את הקובץ ששמו רשום בכתובת **FileName1** כדי לכתוב בו או כדי לקרוא ממנו (הארגומנט m - modify).

**<u>אבל</u>,** אפשר לכתוב אל קובץ או לקרוא ממנו **אך ורק אם הוא קיים.** ברגע זה מקבל המשפט משמעות של **התניה.** אם הקובץ קיים, יופעל המשפט בשורה מתחתיו {setpos ...}. אך אם הקובץ איננו קיים (ואך ורק אם איננו קיים) יופעל המשפט שבהמשך של אותה שורה: {open **FileName1**,w}, שכולל את הארגומנט w (write) - כתוב. אופצית w איננה מתחשבת אם הקובץ קיים או לא - היא פותחת אותו (ולכן תמיד מצליחה). אחרי הפתיחה יהיה גודל הקובץ = 0 בתים. משמעות הדבר היא, שאם היו בו נתונים הם נמחקו.

לסוג זה של התניה חשיבות מרובה ביותר. במקרה שלפנינו נוהל התניה זה מאפשר **פתיחה אוטומטית של קבצים.** במקרים אחרים אפשר להשתמש בנוהל התניה זה בנוסף על נוהלי ההתניה הרבים הקיימים בלוטוס והמקנים לו את כוחו הרב.

המשפט הבא **מציב את הסמן בדיסק** בקובץ שנפתח {setpos ... } במקום בו נרצה לכתוב את הרשומה המעודכנת. הנוהל איננו מסובך. אדגים אותו בתרשים הבא:




אורך הרשומה הוא 32 תווים ומספר הרשומות הוא 3. **שים לב לכך שהספירה בשני המקרים מתחילה ב-0.**

במשפט {setpos ... } שלמעלה כתוב הארגומנט (NN2-1)***Length1**. התא NN2 מכיל את מספר המחלקה שברצוננו לעדכן, מחלקה מספר 2. לכן החישוב (NN2-1)=2-1=1 מפיק את מספר הרשומה בקובץ בדיסק. **Length1=32** הוא

גודל הרשומה. המכפלה של מספר הרשומה בקובץ בגודל הרשומה, נותנת את מספר התו הראשון של רשומת המחלקה (מסומן **x** בציור). זה הוא תו מספר 32=32*(2-1). משפט המקרו הזה מציב **בדיסק** את הסמן. כלומר, כמו שמשפט המקרו {goto} מציב סמן, כך גם המשפט {...setpos} מציב "סמן", אלא שהראשון מציב סמן נראה לעין בגיליון והשני מציב סמן בלתי נראה בדיסק.

עתה נפתח קובץ לקריאה או לכתיבה והסמן הוצב על התו הראשון של הרשומה המבוקשת. עכשיו צריך להורות מה לעשות. נעשה זאת באמצעות המשפט הבא { ... writeln}. משפט זה יכתוב בקובץ שבדיסק, החל מהתו בו מוצב הסמן, את התמליל הכתוב בארגומנט שלו או בכתובת הרשומה בארגומנט. אל הטקסט יתוספו בדיסק שני התווים EOL (End of Line) ו-CR (Carriage Return).

הארגומנט הכתוב במשפט (@index(**Print2**,0,**NN2-1** הוא כתובת "אקטיבית". פונקציה, שנותנת את התפוקה הבאה:

תוכן התא הנמצא ב-**Range** ואשר הקואורדינטות שלו ב-**Range** הזה הן עמודה x ושורה y וכאשר,

- מספר העמודה הראשונה משמאל ב-**Range** הוא x=0
- מספר השורה הראשונה ב-**Range** הוא y=0

התבנית הכללית של הפונקציה:

@index(**Range**,x,y)

קיימת הקבלה מעניינת בין מבנה הפונקציה הזאת לבין מבנה מטריצת נתונים בקובץ ASCII. לדוגמא, נגדיר /rncData1~h3.p5~

A B C D E F G H I J K L M N O PL

1 עמודה בגיליון ⟶ H P

4 מספר העמודה ב Data1 ⟶ 0 1 2 3 4 5 6 7 8

3 מספר השורה ב Data1 ⟶ 0

4 (NN2-1)=(2-1)=1 x

5 ⟵ שורה בגיליון 2

6

7

מכאן קל לראות, שהארגומנט הזה מחזיר את המחרוזת של רשומת מחלקה מספר 2, וכתוצאה מכך נכתבת הרשומה בדיסק. במלים אחרות, פונקציה זו מכתיבה את הפעולות הבאות:

1. גש למטריצה בשם **Range** המוגדרת בגיליון.

2. חפש בתוך המטריצה את העמודה הנמצאת x עמודות מימין לעמודה הראשונה בצד שמאל.

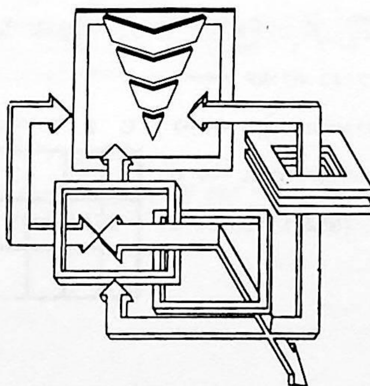
3. בעמודה הזו חפש את השורה הנמצאת y שורות מתחת לשורה הראשונה של המטריצה.

4. החזר את תוכנו של התא שמצאת, יהיה אשר יהיה (value או label).

הפונקציה index@ חשובה ביותר להפעלת תקשורת יעילה בין מטריצת נתונים בגיליון הנמצא על המסך, לבין מטריצת נתונים הנמצאת במבנה ASCII על הדיסק. הזהות במבנה הקואורדינטות תומכת במודולריות בכתיבת Intra-Macro, מכיון ש-NN2 (למשל) מופיע כארגומנט בשני מקומות - בשליפת הנתון מהמטריצה בגיליון ובכתיבתו בדיסק.

עתה עלינו לסגור את הקובץ. המשפט {close} הוא משפט חובה כשמסיימים פעילות בקובץ.

<u>תרגיל:</u> זהו המשך לתרגיל הקודם. הגדר את כל חמשת עמודות המחרוזות בתחום אחד. הפעל את index@ על התחום הזה, **לא** בתוך מקרו. נסה לבחון את התפוקות כשמשנים את מספר העמודה והשורה. חזור על הניסוי כאשר x ו-y הם פונקציות. **תרגיל זה חשוב להמשך.**



## 4.2 כתיבת שדה אחד ברשומה אחת

בסעיף הקודם כתבנו רשומה שלמה אל הדיסק. פעמים רבות יש צורך לעדכן נתון אחד בלבד ברשומה אחת. כגון, סעיף תקציבי מספר 2 במחלקת השיווק. לשם כך יש צורך בנוהל כתיבה אל הדיסק של שדה אחד ברשומה אחת. ה-Intra-Macro הבא מבצע את הפעולה הזאת.

```
               תקציב לחודש ינואר 1990
    ==============================
    סעיף תקציבי           מחלקה
    ============= ===============
    1       2          שם    מספר       רשומות ASCII
    ===== ======== ========== ====  ------------------------------

    1,125  123,654      ייצור   1    | 1125 | 123654 |  ייצור | 1 |
      206    6,333      שיווק   2    |  206 |   6333 |  שיווק | 2 |
      225   24,731     מינהלה   3    |  225 |  24731 | מינהלה | 3 |

                                     |___________ Print1 ____________|

                                                        |  1   0 |
                                                        |  2   6 |
                                                        |  3  15 |
                                                        |  4  26 |
                                                         Fields1

      \c {getnumber Msg1,NN2}~{getnumber Msg2,NN3}~
         {open FileName1,m}{open FileName1,w}
         {setpos (NN2-1)*Length1+NN4}
         {write @index(Print1,NN3-1,NN2-1)}
         {close}
         {quit}
     NN2        <------- מספר המחלקה
     NN3        <--- מספר סעיף תקציבי
     NN4 @vlookup(NN3,Fields1,1) <--- תו ראשון בשדה NN3 ברשומה
 Length1 32     <--------------- FileName1 גודל רשומה ב
FileName1 +Drive1&Dir1&File1&Ext1
  Drive1 c:\
    Dir1 Ran\WS_5\Data1\
   File1 01_90
    Ext1 .Ran{return}
    Msg1       (1..3) הקש את מספר המחלקה -->
    Msg2     (1..2) הקש את מספר סעיף התקציב -->
```

הבה נבחן את פיסקת המקרו c\ אשר רשומה בשורה הראשונה.

עד כה התייחסנו במפורש לשני מרכיבים ראשיים של מערך המידע:

1. המקומות בהם שומרים את המידע והמקומות בהם מבצעים פעולות על המידע. דהיינו, בלשון פשוטה, הדיסק והמסך.

2. נוהלים ושיטות. כלומר, Meta-Macro ו-Intra-Macro.

ישנו מרכיב שלישי וחשוב לא פחות מהשניים האחרים. מרכיב שעד כה ההתייחסות אליו לא היתה מפורשת:

3. האדם שיושב אל מול המסך.

שני משפטי המקרו הראשונים בפיסקה זו יוצרים קשר נוח וזמין בין המפעיל לבין המחשב. הבה נראה את סדר הפעולות שיתן את התפוקה המבוקשת - העתקת נתון מתא בגיליון (שהוא שדה אחד ברשומה אחת) אל קובץ ASCII בדיסק.

**המפעיל:**

1. עדכן את הסכום בסעיף התקציבי של המחלקה.

2. רשום את מספר המחלקה לעדכון בכתובת (תא) כל שהוא בגיליון.

3. רשום את מספר הסעיף התקציבי לעדכון בכתובת (תא) אחר כל שהוא בגיליון.

כאן מתבקש המפעיל לעדכן את הנתון בגיליון ולרשום אחר כך את מספר המחלקה והסעיף בשתי כתובות בגיליון. כדי לחסוך ממנו את החיפוש אחר הכתובות הללו וכדי להפיק תקשורת אדם-מחשב במירב הנוחיות נפעיל את המקרו הבא:

```
\c {getnumber Msg1,NN2}~{getnumber Msg2,NN3}~
```

עם הפעלתו של משפט זה יתבצעו הדברים הבאים:

1. משפט המקרו הראשון בשורה זו...

1. שולף את ההודעה הכתובה בכתובת **Msg1** וכותב אותה בשורת הדו-שיח השנייה (השורה האמצעית מבין שלושת השורות שמעל הסרגל העליון)

2. ממתין שנקליד את התשובה להודעה (מספר המחלקה) ונקיש ENTER.

3. אחרי שהקשנו ENTER, כותב את התשובה שהקלדנו בכתובת NN2.

2. משפט המקרו השני...

1. שולף את ההודעה הכתובה בכתובת **Msg2** וכותב אותה בשורת הדו-שיח השניה.

2. ממתין שנקליד את התשובה להודעה (מספר הסעיף) ונקיש ENTER.

3. אחרי שהקשנו ENTER, כותב את התשובה שהקלדנו בכתובת NN3.

כאן הסתיימה האינטראקציה בין האדם והמחשב והפעולות הבאות מתבצעות ע"י המחשב בלבד.

<u>המחשב:</u>

1. פתח קובץ {open

2. אשר שמו רשום בכתובת **FileName1**

3. לקריאה או לכתיבה {m,

4. ואם הקובץ איננו קיים, אזי הקם אותו לכתיבה

{open **FileName1**,w}

5. הצב את הסמן **בדיסק**

{setpos

6. על התו הראשון של שדה הסעיף התקציבי המבוקש ברשומת המחלקה המבוקשת

**(NN2-1)*Length1+NN4**

<u>סעיף זה דורש הסבר:</u>

המחלקות ממוספרות מ-1 ואילך. הרשומות בקובץ ASCII ממוספרות מ-0 ואילך. לכן, מספר הרשומה בקובץ ASCII הינו מספר המחלקה פחות אחד. כאשר כופלים את התוצאה בגודל הרשומה, מקבלים בדיוק את התו הראשון משמאל בשדה הראשון של רשומת המחלקה המבוקשת בדיסק. מספר התו הזה מתקבל על ידי המכפלה **(NN2-1)*Length1**, שהרי בכתובת **NN2** רשום מספר המחלקה ובכתובת **Length1** - מספר התווים ברשומה.

לאחר שחושב מספר התו הראשון של **הרשומה,** צריך למצוא את מספר התו הראשון **של השדה** המבוקש. לשם כך אפשר לבצע נוהלים שונים. כאן נשתמש בפונקציה **@vlookup**.

<u>נדגים את מהלך העבודה:</u>

מבצעים **/rncFields1~** כך שיכיל שתי עמודות וארבע שורות. בעמודה הראשונה בכל שורה כתוב המספר הסידורי של השדה (בדוגמא זו - מ-1 ועד 4). בעמודה השנייה של כל שורה כתוב מספר התו הראשון של השדה (נקח זאת מטופס מבנה רשומה שהוכן מראש). עכשיו, בכתובת NN4 כותבים את הפונקציה הבאה:

```
@vlookup(NN3,Fields1,1)
```

| | |
|---|---|
| 1 | 0 |
| 2 | 6 |
| 3 | 15 |
| 4 | 26 |

**Fields1**

מכיון שבכתובת NN3 מופיע מספר הסעיף התקציבי, מחזירה הפונקציה הזאת את מספר התו הראשון של שדה הסעיף התקציבי המבוקש. כל שנותר לעשות הוא להוסיף את המספר הזה למספר התו הראשון ברשומה:

```
(NN2-1)*Length1+NN4
```

**7.** סיים משפט מקרו זה כראוי.

```
}
```

**8.** במקום הזה בדיסק עליך לכתוב

```
{write
```

**9.** את...

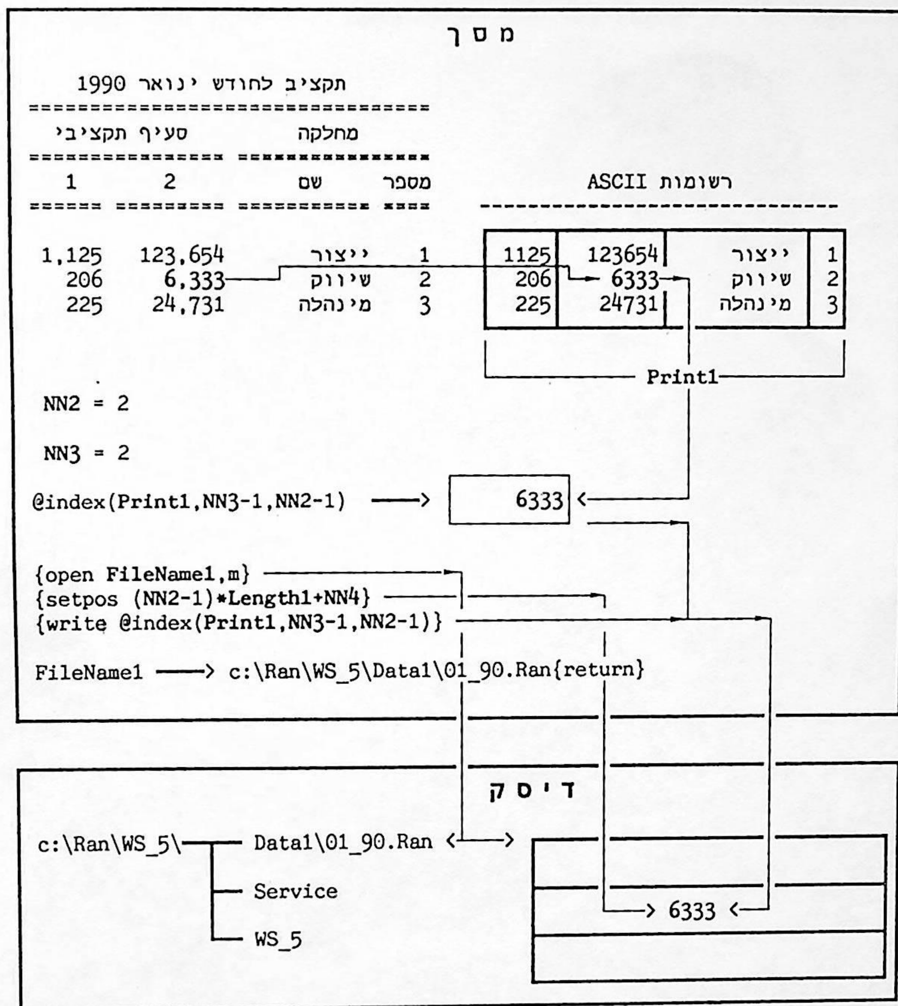
```
@index(Print1,NN3-1,NN2-1)
```

אנחנו כבר יודעים, שהפונקציה הזאת מפיקה את תוכן התא בגיליון שכתובתו: "השטח ששמו **Print1**, עמודה מספר NN3-1 משמאל, שורה מספר NN2-1 מלמעלה". התא הזה הרי מכיל את **משוואת השדה** של הנתון שעודכן. ולפיכך, כל שנותר הוא ...

**10.** סיים משפט מקרו זה כראוי

```
}
```

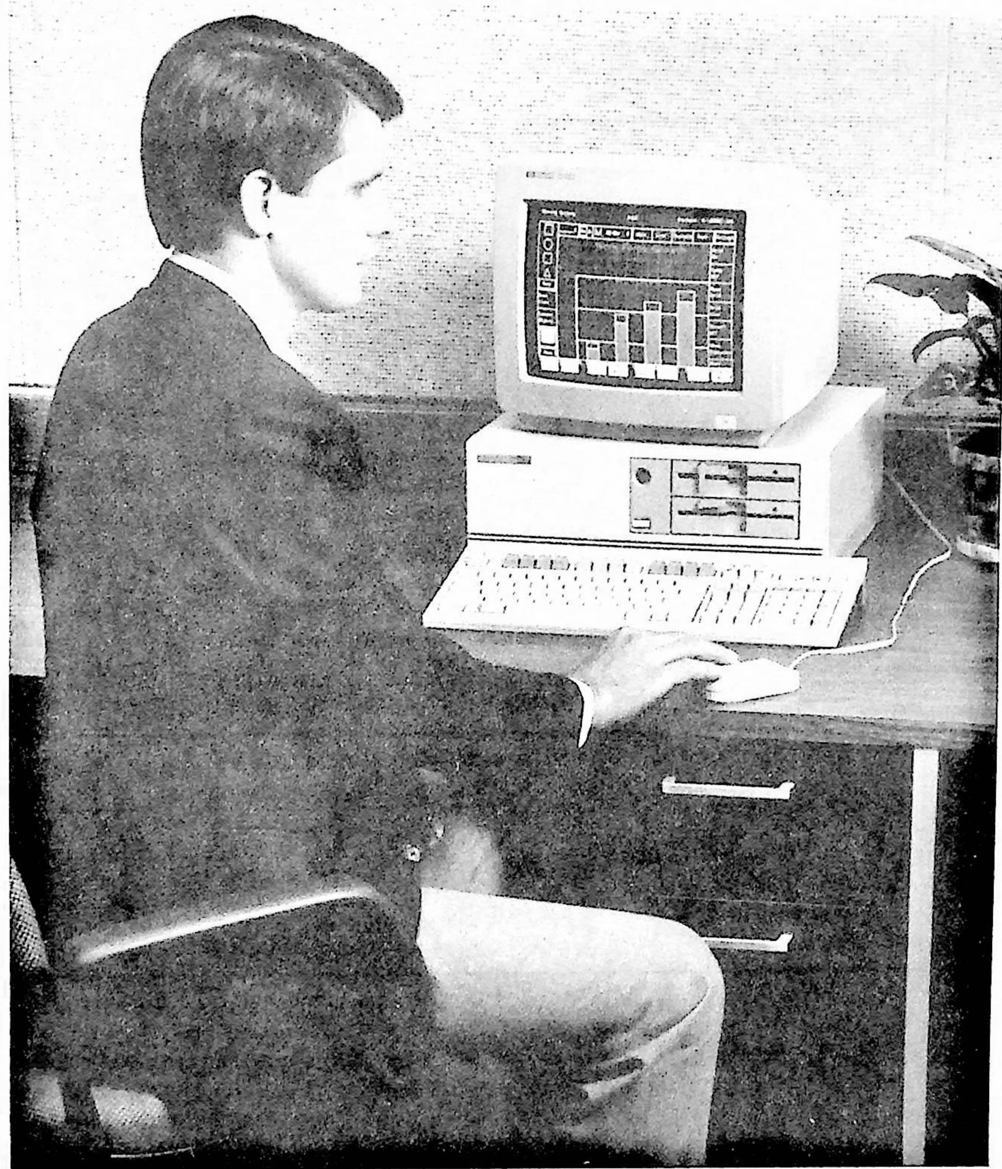
עדכון הדיסק יתבצע כפי שבקשנו. לשם המחשה נתבונן בתרשים הבא.




**.11** סיים פיסקת מקרו זו כראוי

```
{close}
{quit}
```

כאן הסתיימה תגובת המחשב.



## 4.3 כתיבת רצף רשומות

בדרך כלל כותבים את כל הרשומות ברצף אל הדיסק. בחלק מהמקרים הללו אפשר להשתמש בפקודת התפריט pf/, אך בהזדמנויות רבות נרצה להפעיל את המקרו { ... open} בלולאה. לולאה כזאת ניתן ליצור בשני אופנים:

1. על ידי { ... let}
2. על ידי { ... for}

בפרק זה נבחן את שני האופנים הללו: תחילה את הלולאה { ... let} שהיא פשוטה ללימוד, ואחר כך נציג את הלולאה { ... for} אשר יעילה ומהירה יותר ממנה.

תקציב לחודש ינואר 1990

| מספר | שם | סעיף תקציבי 2 | סעיף תקציבי 1 |
|---|---|---|---|
| 1 | ייצור | 123,654 | 1,125 |
| 2 | שיווק | 6,333 | 206 |
| 3 | מינהלה | 24,731 | 225 |

רשומות ASCII

| | | | |
|---|---|---|---|
| 1 | ייצור | 123654 | 1125 |
| 2 | שיווק | 6333 | 206 |
| 3 | מינהלה | 24731 | 225 |

Print1

```
\a {let NN1,0}~{open FileName1,m}{open FileName1,w}
   {setpos 0}
 1 {writeln @index(Print1,0,NN1)}
   {let NN1,NN1+1}~{branch 3}
 2 {close}
   {quit}
 3 {if NN1<@rows(Print1)}{branch 1}
   {branch 2}
NN1                   <— מונה פעימות
FileName1 +Drive1&Dir1&File1&Ext1
   Drive1 c:\
     Dir1 Ran\WS_5\Data1\
    File1 01_90
     Ext1 .Ran{return}
```

לוח זה וחלקים רבים ממקרו a\ שלמעלה כבר מוכרים לנו מדוגמאות ותרגילים קודמים.

תחום **Print1** הוגדר על עמודה אחת ומכיל את שלוש **משוואות הרשומה** של הנתונים בלוח שמשמאלו. באיזור המקרו מגדירים ~rncNN1/ שישמש מונה פעימות. הבה נעבור על פיסקת המקרו **a**\ לפי סדר הפעולות:

1. איפוס מונה הפעימות <— NN1=0

2. פתיחת **FileName1** עם אופצית **m**

3. הצבת הסמן **בדיסק** בתחילת הקובץ

4. העתקת' הרשומה הראשונה מ-**Print1** אל **FileName1**

@index(Print1,0,0) <— @index(Print1,0,NN1)

5. מגדילים את המונה ~{let NN1,NN1+1} באחד

6. מתפצלים אל פיסקת ההתניה **3:**

אם **NN1** קטן ממספר הרשומות ב-**Print1** מתפצלים למשפט **1** שיעתיק את הרשומה השנייה מ-**Print1** אל **FileName1**_
המשפט {writeln...} כותב את הרשומה בדיסק ומציב מיד את הסמן בדיסק על התו הראשון של הרשומה הבאה בתור_

אחרת, מתפצלים ל-**2**, סוגרים את הקובץ ומסיימים את הפיסקה.

__תרגיל משולב:__

למדנו לכתוב מהגיליון אל הדיסק:

רשומה אחת **b**\

שדה אחד ברשומה אחת **c**\

רצף של רשומות **a**\

כתוב והפעל Intra-Macro שיכלול את כל שלוש המטלות הללו במערך מודולרי אחד, המופעל אוטומטית ע"י תפריט פקודות במבנה {menu...}.

## 4.4 כתיבת דיווח אוטומטי

בניהול תקין יש חשיבות רבה לדיווח אוטומטי של מועדי ביצוע של פעולות שונות למעקב החלטות. במיוחד נודעת חשיבות למועדי עדכון נתונים. ניתן לדמות מסד נתונים המכיל גם שדה ובו תאריך עדכון אחרון של נתונים הנמצאים בשדה שכן. שדה תאריך העדכון **סמוי** עבור מי שקולט את הנתונים. כלומר, בעת הקשת הנתונים המפעיל יראה את כותרות העמודות לנתונים המוקלדים בלבד כגון, "סעיף תקציבי 1 - ביצוע". אבל בקובץ ה-ASCII בדיסק שיועד לשמור את הנתונים יש משבצת מוכנה לקלוט את תאריך ההקלדה. תאריך זה ניתן לשלב כפונקציה בתוך **משוואת השדה** של הנתון הגלוי (למשל, ביצוע בש"ח - סעיף תקציבי 1). התוצאה תהיה:

הרשומה שתיכתב בדיסק תכלול שילוב של נתונים גלויים עם נתונים סמויים.

נתונים סמויים אינם מוגבלים לשדה התאריך, אלא יכולים לייצג כל ערך רצוי. הבוחן הוא שיש לשמור את הערך הזה, אך הוא אינו נדרש בעת הקלדת הנתונים. לעתים גם רוצים לשמור אותו חסוי מהמשתמש, כמו למשל, ההפרש בין התקציב המתוכנן לבין הביצוע.

הדוגמא הבאה מתארת ארגון בן שני מפעלים, **א** ו-**ב**, וארבע מחלקות מרכזיות - שיווק, תובלה, הדרכה, ומינהלה. לכל אחת מהיחידות הללו יש תקציב מתוכנן ונוהל מעקב אחרי ביצוע.

ברגע מסוים נראים הנתונים הללו כפי שהם מוצגים בלוח הבא. לוח זה מאפשר עדכון כל נתון ושמירתו בדיסק. עד כאן המרכיבים הגלויים.

נאמר שכל פעם שמעדכנים נתון בדיסק רוצים לעדכן בו-זמנית גם את תאריך העדכון לשם מעקב וביקורת. הדרך הטובה ביותר לקלוט נתון זה תהיה, אם הוא יירשם באופן אוטומטי לגמרי ללא מעורבות המשתמש.

**סעיף תקציבי**

| | יחידה | 6 | | 5 | | 4 | | 3 | | 2 | | 1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| # | שם | תכנון | ביצוע | תכנון | ביצוע | תכנ. | ביצ. | תכנ. | ביצ. | תכנ. | ביצ. | תכנון | ביצוע |
| 1 | מפעל א | 55000 | 49500 | 27500 | 24750 | | | 6875 | 6188 | 3438 | 3094 | 34380 | 30940 |
| 2 | מפעל ב | 23500 | 25850 | 11750 | 12925 | | | 2938 | 3232 | 1469 | 1616 | 14690 | 16160 |
| 3 | שיווק | 15250 | 16775 | 7625 | 8388 | | | | | 954 | 1049 | | |
| 4 | תובלה | 12000 | 10800 | 6000 | 5400 | 3000 | 2700 | 1500 | 1350 | 750 | 675 | 7500 | 6750 |
| 5 | הדרכה | 4000 | 3600 | 2000 | 1800 | 1000 | 900 | 500 | 450 | | | 2500 | 2250 |
| 6 | מינהלה | 7000 | 6300 | 3500 | 3150 | 1750 | 1575 | 875 | 788 | | | 4380 | 3940 |

תחילה מכינים את מבנה הרשומה כך, שתכלול שדה לקליטה של התאריך עבור כל נתון תקציבי, כפי שמודגם בטופס מבנה הרשומה הבא.

**טופס מבנה רשומה**

**תקציבי היחידות - תכנון וביצוע**

שם הקובץ: c:\Ran\WS_5\Data2\T_1.Ran

אורך רשומה = 213 = 2 + 211

| הערות | מספר | שדה - תיאור | | | y) | @mid(x, |
|---|---|---|---|---|---|---|
| לכל סעיף תקציבי יש | 1 | סעיף 1 | ביצוע | תאריך עדכון אחרון | 9 | 0 |
| 32 תווים ב 4 שדות: | 2 | | | ש"ח | 7 | 9 |
| | 3 | | תכנון | תאריך עדכון אחרון | 9 | 16 |
| ביצוע: תאריך עדכון | 4 | | | ש"ח | 7 | 25 |
| תקציב בש"ח | 5 | סעיף 2 | ביצוע | תאריך עדכון אחרון | 9 | 32 |
| תכנון: תאריך עדכון | 6 | | | ש"ח | 7 | 41 |
| תקציב בש"ח | 7 | | תכנון | תאריך עדכון אחרון | 9 | 48 |
| | 8 | | | ש"ח | 7 | 57 |
| לשדה תאריך - 9 תווים | 9 | סעיף 3 | ביצוע | תאריך עדכון אחרון | 9 | 64 |
| לשדה ש"ח - 7 | 10 | | | ש"ח | 7 | 73 |
| | 11 | | תכנון | תאריך עדכון אחרון | 9 | 80 |
| | 12 | | | ש"ח | 7 | 89 |
| | 13 | סעיף 4 | ביצוע | תאריך עדכון אחרון | 9 | 96 |
| | 14 | | | ש"ח | 7 | 105 |
| | 15 | | תכנון | תאריך עדכון אחרון | 9 | 112 |
| | 16 | | | ש"ח | 7 | 121 |
| | 17 | סעיף 5 | ביצוע | תאריך עדכון אחרון | 9 | 128 |
| | 18 | | | ש"ח | 7 | 137 |
| | 19 | | תכנון | תאריך עדכון אחרון | 9 | 144 |
| | 20 | | | ש"ח | 7 | 153 |
| | 21 | סעיף 6 | ביצוע | תאריך עדכון אחרון | 9 | 160 |
| | 22 | | | ש"ח | 7 | 169 |
| | 23 | | תכנון | תאריך עדכון אחרון | 9 | 176 |
| | 24 | | | ש"ח | 7 | 185 |
| | 25 | יחידה ומספרה | | | 19 | 192 |

עכשיו צריך להחליט באיזו צורה לשמור את התאריך. אפשר לשמור את חמש הספרות של לוח התאריכים בלוטוס, או בכל מבנה אחר. בחרתי להדגים כאן ביצוע טרנספורמציה של התאריך לתווי ASCII במבנה של "dd.mm.yy". אפשר כמובן לבחור כל מבנה אחר רצוי.

```
+" "&@repeat(" ",2-@length(@string(@day(@now),0)))&@string(@day(@now),0)&"."
@repeat(" ",2-@length(@string(@month(@now),0)))&@string(@month(@now),0)&"."
@repeat(" ",2-@length(@string(@year(@now),0)))&@string(@year(@now),0)
+Day1&Month1&Year1

   Date1   7.12.90    <-
    Year1 90          <-
  Month1 12.          <-
     Day1   7.        <-
```

התאריך שיישמר מופק מהמשוואה ב-**Date1** שהיא למעשה, **רשומת שדה**. אבל גם משוואה זו מורכבת מ**רשומות שדה** משלה, כאשר **Day1**, **Month1**, **Year1** מכילים משוואות שדה בהתאמה.

עכשיו מכינים את נוהל השמירה.

| | A | B |
|---|---|---|
| 1 | | |
| 2 | | 1 |
| 3 | ——— | ——— ... |
| 4 | ביצוע | תכנון |
| 5 | ——— | ——— |
| 6 | | |
| 7 | 30940 | 34380 |
| 8 | 16160 | 14690 |
| 9 | 1049 | 954 |
| 10 | 6750 | 7500 |
| 11 | 2250 | 2500 |
| 12 | 3940 | 4380 |

**Print1**

```
+Date1&@repeat(" ",7-@length(@string(a7,0)))&@string(a7,0)
```

קיבלנו **משוואת שדה משולבת**. חלקה הראשון הוא התאריך והחלק **השני** הנתון בש"ח, כששניהם כבר במבנה תואם לשמירה ברשומה בדיסק. **המשוואה** המשולבת הזאת **בשלמותה** כתובה (בדוגמא שלפנינו) בתא השמאלי העליון **של Print1**. המשוואה הזו מחזירה:

**Print1**

| | |
|---|---|
| 30940  7.12.90 | |
| | |

כעת, המשפט {write @index(Print1,x,y)} יעתיק אל הדיסק את הנתון בש"ח ואת התאריך הנוכחי, שהוא תאריך הקלדת העדכון.

# 5. שליפה מהדיסק

## 5.1 שליפת שדה ברשומה

הבה נתאר לעצמנו מנהל בכיר בעת דיון תקציבי. לידו מחשב והוא מבקש לקבל **מיד** את התקציב המתוכנן ליחידה 2 בסעיף תקציבי 2, את דוח הביצוע, את ההפרש ביניהם, החריגה מהתקציב והזמן שעבר (בימים ובשבועות) מהרגע שבו עודכנו נתוני התכנון והביצוע לאחרונה.

השלב הראשון בכתיבת Intra-Macro לביצוע מטלה זו דורש חשיבה **על המשתמש.** לפיכך נתכנן תחילה את המסך כפי שיראה סופית לאחר שיישלפו הנתונים המבוקשים. הרי דוגמא של מסך כזה:

```
Alt z ... תפריט

       2 ...מספר          מפעל ב ... יחידה

       2 ...סעיף תקציבי מספר
==============================
  זמן שעבר      תאריך                   7.12.90 ... תאריך נוכחי
============    עדכון
 שבועות ימים    אחרון      ש"ח
===== ====== ======== ========

 239      34 12. 4.90   1,469 ... תכנון

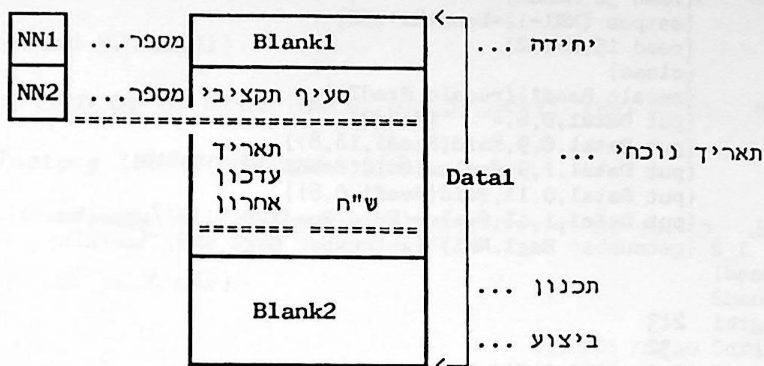
  53       8 15.10.90   1,616 ... ביצוע
===================== --------
                  ש"ח   (147) ... תכנון פחות ביצוע

           % חריגה    -10.01
                     ========
```

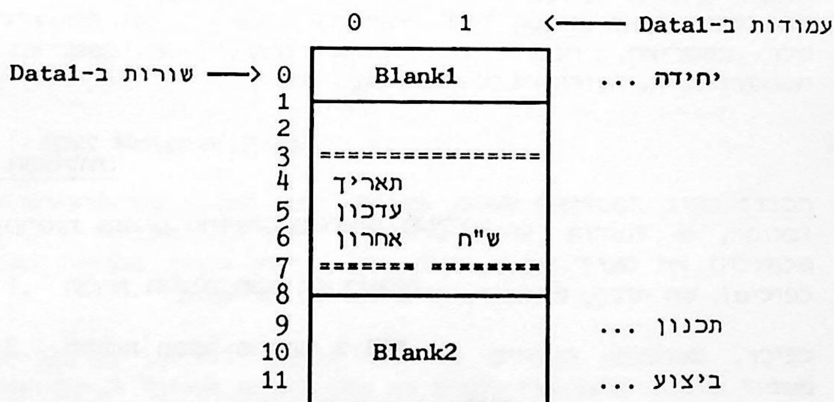
הדבר הראשון שבולט לעין הוא שחלף זמן רב (8 שבועות) מאז שעודכנו נתוני הביצוע לאחרונה. המנהל יכול מיד לדרוש עדכון, ומשיודיעו לו שהעדכון בוצע ישלוף מחדש את הנתונים. אפשר כמובן גם לבצע עדכון נתונים אוטומטי שוטף, וזאת גם כאשר מקור הנתונים בתוכנת הנהלת חשבונות או מסד נתונים כל שהוא.

נלמד את המסך הזה בשלבים. תחילה נלמד כיצד לשלוף נתונים מקובץ ASCII אל המסך. נמשיך בהצבה של כל אחד במקום המיועד לו על המסך. ולבסוף - נציג את המשוואות להפקת החישובים.

נגדיר על **מסך הקלט** מספר תחומים:




נציג את מבנה התחומים בצורה סכימתית:



המקרו לפתרון בעיה זו:

```
      1_1 /wgpd{1_2}{blank Blank1}{blank Blank2}~{open FileName1,r}
          {setpos (NN1-1)*Length1+(NN2-1)*Length2}
          {read 32,Read1}
          {setpos (NN1-1)*Length1+192}
          {read 15,Read2}
          {close}
          {recalc Read1}{recalc Read2}~
          {put Data1,0,0,+"  "&Read2}~
          {put Data1,0,9,@mid(Read1,16,8)}
          {put Data1,1,9,@value(@mid(Read1,25,7))}
          {put Data1,0,11,@mid(Read1,0,8)}
          {put Data1,1,11,@value(@mid(Read1,9,7))}~/wgpe{home}{quit}
      1_2 {getnumber Msg1,NN1}~{getnumber Msg2,NN2}~{return}
    Read1
    Read2
  Length1  213
  Length2   32
FileName1 +Drive1&Dir1&File1&Ext1
   Drive1 c:\
     Dir1 Ran\WS_5\Data2\
    File1 T_1
     Ext1 .Ran{return}
     Msg1          1..6  מספר המחלקה ->
     Msg2      1..6  מספר הסעיף התקציבי ->
```

<u>סדר הפעולות:</u>

1. קריאת נתונים מהדיסק והצבתם בגיליון:

1. הסרת ההגנה מכתיבה בגיליון.

2. קליטת מספר מחלקה ב-NN1.

3. קליטת מספר הסעיף התקציבי ב-NN2.

4. ניקוי האיזורים שיקבלו את הנתונים.

5. פתיחת הקובץ לקריאה בלבד: {open FileName1,r}.

6. הצבת הסמן בדיסק לשליפת ארבעת השדות של הסעיף התקציבי.

Length1 = 213 הוא אורך הרשומה.

Length2 = 32 הוא מספר התווים בארבעת השדות של כל סעיף תקציבי.

ומכאן, {setpos (NN1-1)*Length1+(NN2-1)*Length2}

מציב את הסמן בדיסק על התו הראשון של השדה הראשון של הסעיף התקציבי המבוקש ברשומת היחידה המבוקשת.

7. קריאת 32 התווים הבאים מהדיסק והצבתם ב-**Read1** על ידי:

```
{read 32,Read1}
```

8. הצבת הסמן בדיסק על התו הראשון בשדה עם שם היחידה:

```
{setpos (NN1-1)*Length1+192
```

9. קריאת שם היחידה מהדיסק והצבתו ב-**Read2** בגיליון:

```
{read 15,Read2}
```

10. סגירת הקובץ.

11. ביצוע {recalc} כדי שהקריאה מהדיסק תראה בגיליון.

2. הצבת כל נתון במקום המיועד לו על המסך:

המקרו { ... put} הוא ההופכי של הפונקציה @index. בעוד שהפונקציה הזאת <u>שולפת</u> נתון מתוך תא במטריצה לפי הקואורדינטות שלו במטריצה, המקרו הזה <u>כותב</u> נתון בתא במטריצה לפי הקואורדינטות. הצורה הכללית של מקרו זה:

```
{put Range,x,y,z}
```

<u>הסבר:</u> כתוב במטריצה ששמה **Range**, בתא הנמצא בקואורדינטות x,y (כלומר, x עמודות ימינה מהעמודה הראשונה משמאל ו-y שורות מלמעלה) את הערך של z. הערך של z יכול להיות מחרוזת (בגרשיים כפולים), שם תחום, פונקציה, או שילוב של כל אלה.

לפיכך, משנשלפו הנתונים מהדיסק ונכתבו בגיליון, נשתמש במספר משפטי { ... put} כדי להציב כל נתון למקום המיועד לו על המסך.

בעת כתיבת המקרו כותבים את משפט ה-{ ... put} הראשון, מבצעים עליו @repeat ומשנים אחר כך רק את ערכי x ו-y.

בסיום מחזירים את ההגנה על הגיליון: /wgpe

**תרגיל:** 1. כתוב מקרו בשם z\ שיפעיל תפריט פקודות למטלות הבאות:

1. שליפת נתון - הפעלת הפיסקה 1_1.

2. כתוב פיסקת מקרו 3_1 שתבצע את המטלה הבאה: נאמר ששלפנו ערכים, שינינו אותם ואנו רוצים לשמור את השינויים בדיסק.
רמז: הערכים של NN1 ו-NN2 כבר ידועים.

3. יציאה לתוכן העניינים - m\

2. כתוב את כל המשוואות שישלימו את המידע המבוקש להקרנה במסך זה.

## 5.2 שליפת רצף רשומות

במקרים רבים יש צורך לשלוף רשומות ברצף. הבה נבחן את הדוגמא הבאה.

הקמנו בלוטוס יישום ששומר נתונים בקובץ ASCII בגודל אינסופי. כלומר, במהלך הפעלת היישום מתוספות רשומות לקובץ בנוהל {open FileName1,a} מבלי שיהיה צורך לייבא את הקובץ לשם כך אל הגיליון. הקובץ הלך וגדל ואיננו יודעים כמה רשומות יש בו. עכשיו רוצים לשלוף מתוכו מספר רשומות ברצף, דהיינו, לשלוף אל הגיליון רשומות זו אחר זו, החל מרשומה מסוימת שמספרה ידוע לנו וסיום לאחר מספר מוגדר של רשומות. את הרשומות הנשלפות רוצים להציב במקום כל שהוא בגיליון ולבצע עליהן עבודה כל שהיא.

פעולה כזאת נדרשת למשל, כשרוצים לעבד נתונים מקבוצת רשומות כדי להכין ולהדפיס דוח, או כשרוצים להדפיס מדבקות לחלק מהאוכלוסיה שבקובץ כתובות. לצורך ההסבר נשתמש בדוגמת המדבקות.

נתון קובץ ASCII שברשומותיו שמות וכתובות וזה הוא המבנה שלו:

| | | | | | |
|---|---|---|---|---|---|
| אבן-טובישראל | משרד התיירות | ג'ורג' 24 | ירושלים | 91708 |
| בן-דודמיכל | מפעלי ים המלח בע"מ | 75 | באר שבע | 84100 |
| גוגנהייםדניאל | ציון חברה לביטוח בע"מ | נשרים 3 | ירושלים | 95461 |

המבנה הסכימתי של הרשומה:

| כתובת | | | | שם | |
|---|---|---|---|---|---|
| מיקוד | ישוב | רחוב ומספר | שם הארגון | פרטי | משפחה |
| | | | | | |

גוגנהייסדניאל ציון חברה לביטוח בע"מ נשרים 3 ירושלים 95461

| 5 | 12 | 19 | גודל השדה = 25 תווים | 7 | 8

0 5 17 36 = מספר תו ראשון 61 68

נציג זאת בצורה מסודרת יותר:

**טופס מבנה רשומה**

**רשימת כתובות למדבקות**

שם הקובץ: c:\Ran\WS_5\Data2\1_1.Ran

אורך רשומה = 78 = 2 + 76

| הערות | שדה מספר | תיאור | @mid(x, | y) |
|---|---|---|---|---|
| | 1 | מיקוד | 0 | 5 |
| | 2 | ישוב | 5 | 12 |
| | 3 | רחוב ומספר | 17 | 19 |
| | 4 | שם הארגון | 36 | 25 |
| | 5 | שם: פרטי | 61 | 7 |
| | 6 | משפחה | 68 | 8 |

הדבר הבא לעשות הוא להקים את פורמט הפלט. בדוגמא שלנו יהיה זה מבנה הכתובת, בדיוק כפי שתודפס על המדבקה.

במדבקה בגודל סטנדרטי של 70x35 מ"מ אפשר להדפיס 8 שורות ב-setup של 014\ עם שורת רווח בין מדבקה למדבקה. את ההדפסה נבצע כך:

נקים **Print1** בגיליון על עמודה אחת שרוחבה יהיה כמספר התווים שאפשר להדפיס בשורה אחת במדבקה + 1, נאמר 28=1+27. נפרוס את התחום הזה על פני 8 שורות.

| | A | B | C |
|---|---|---|---|
| 1 | | | |
| 2 | | 0 | |
| 3 | | | |
| 4 | 0 | | |
| 5 | 1 | | |
| 6 | 2 | | |
| 7 | 3 | Print1 | |
| 8 | 4 | | |
| 9 | 5 | | |
| 10 | 6 | | |
| 11 | 7 | | |
| 12 | | | |
| 13 | | | |

פיסקאות המקרו שיבצעו את הדפסת המדבקות:

```
\z       1 {open FileName1,r}
           {filesize Stop1}
           {let Stop1,Stop1/Length1}~{for Counter1,Start1,Stop1,1,2}~{3}
         2 {readln Read1}
           {put Print1,0,0,@mid(Read1,61,7)&" "&@mid(Read1,68,8)}   שם הנמען
           {put Print1,0,2,@mid(Read1,36,25)}                        שם הארגון
           {put Print1,0,4,@mid(Read1,17,19)}                        רחוב ומספר
           {put Print1,0,5,@mid(Read1,5,12)}                         יישוב
           {put Print1,0,6,@mid(Read1,0,5)}                          מיקוד
           ~{4}{return}
         3 {close}
           {quit}
         4 /rlrPrint1~/pprPrint1~omnouqglq{blank Print1}~{return}
Counter1
  Start1       0
   Stop1
   Read1
 Length1      78
FileName1 c:\Ran\WS_5\Data2\1_1.Ran{return}     <—— +Drive1&Dir1&File1&Ext1
  Drive1 c:\
    Dir1 Ran\WS_5\Data2\
   File1 1_1
    Ext1 .Ran{return}
```

חיבור Intra-Macro זה מפיק מספר אין סופי של מדבקות בנוהל הבא: שליפת רשומה, הצבתה ב-**Print1** במבנה תואם למדבקה, הדפסת המדבקה, דילוג על שורת הרווח בין המדבקות, שליפת הרשומה הבאה וחוזר חלילה. העבודה מסתיימת אוטומטית אחרי הדפסת הכתובת האחרונה.

מהירות השליפה והחישוב גבוהה ביותר. ההדפסה מתחילה מיד עם הפעלת המקרו ונמשכת בשטף וללא כל המתנה בין המדבקות.

בדוגמא זו נוצר רצף מדבקות שאורכו, במספר מדבקות רצופות, כמספר הכתובות. אפשר לקבל מדבקות בדפים, אשר בכל אחד מהם מספר שורות ומספר עמודות. איך נדפיס מדבקות אלו?

את הפתרון נשאיר לתרגיל, אך תחילה לסדר הפעולות של z\:

**1. סדר הפעולות עד להפעלת { ... for}:**

1. פתח את הקובץ לקריאה בלבד:

{open **FileName1**,r}

2. חשב את גודל הקובץ וכתוב אותו ב-**Stop1**:

{filesize **Stop1**}

3. חשב את מספר הרשומות בקובץ וכתוב את התוצאה ב-**Stop1**

{let Stop1,**Stop1/Length1**}~

לצורך הדוגמא, נאמר שקיבלנו את המספר 12.

4. בצע את הלולאה:

{for **Counter1**,**Start1**,**Stop1**,1,2}~{3}

המקרו {...for} שולט במשפט של מקרו שירות, ששמו הוא הארגומנט החמישי של {...for}. משפט שירות זה מופעל פעם אחר פעם, וכל הפעלה נרשמת במונה הפעימות ששמו הוא הארגומנט הראשון של {...for}. הפעימות נפסקות כאשר מונה הפעימות מראה מספר גדול יותר מהמספר הרשום בכתובת, שהיא הארגומנט השלישי של {...for}.

בשלב קודם בפרק זה הקמנו לולאה כזאת על ידי שימוש ב-{...let}. העיקרון דומה, אם כי ב-{...for} יש יותר אופציות לשליטה בלולאה, העבודה יותר אוטומטית והביצוע הרבה יותר מהיר.

כדי "לדעת" מתי לעצור, צריך לעקוב אחרי הביצוע וכאשר דבר מה במהלך הביצוע עונה על קריטריון נתון נפסק הביצוע. לשם כך מגדירים **מונה** "פעימות". כל אימת שמשפט השירות שהופעל תחת { ... for} מסיים ביצוע **פעם** אחת, משתנה המספר במונה. אפשר לגרום שהמספר במונה יגדל כל **פעם** ב-1, ב-2, ב-10, או בכל **מדרגה** שרוצים. בדוגמא שלנו נמצא **מונה הפעימות** ב-**Counter1**. כותבים את כתובתו של מונה הפעימות כארגומנט הראשון { ... ,for Counter1}. את **המדרגה** בספירת **הפעימות** כותבים בארגומנט הרביעי. כאן בחרתי לא לכתוב שם תחום בארגומנט, אלא את המדרגה עצמה: {for **Counter1**,...,...,1,...}.

הארגומנט השני מתחיל את **מונה הפעימות** ממספר כל שהוא. נוכל להתחיל את **המונה** באפס, ב-1, או במספר אחר כל שהוא. למספר הזה חשיבות רבה.

ניתן לכתוב **Counter1** גם כארגומנט ב-@index, ב-{put ... }, או בכל מקום אחר. כך אפשר לפתור ללא קושי את הבעיה שהצגנו להדפסת דף המדבקות (עוד נדון בכך בהמשך). בדוגמא הנוכחית נכתוב 0.

הארגומנט השלישי מורה מתי להפסיק. כל זמן שהמספר **במונה הפעימות** שווה או קטן מהמספר הכתוב בכתובת **Stop1**, מופעל משפט השירות שוב ושוב. כאשר המספר ב**מונה הפעימות** גדול יותר מהמספר ב-**Stop1** נפסק הביצוע והמחשב "ממשיך לקרוא" את מלות המקרו שכתובות מימין ל-{for ... } ובאותה שורה

{for **Counter1**,**Start1**,**Stop1**,1,...}

לבסוף צריך לכתוב בארגומנט החמישי את הכתובת שבה נמצא משפט השירות:

{for **Counter1**,**Start1**,**Stop1**,1,2}~

המצב ההתחלתי להפעלת הלולאה הוא, אם כן,

מונה הפעימות מאופס, הפסקת ביצוע כאשר המונה יראה 12, המונה יגדל כל פעם ב-1 ומשפט השירות הוא 2.

## 2. סדר הפעולות של משפט השירות תחת { ... for}:

1. קריאת הרשומה הראשונה מהדיסק

{readln **Read1**}

עם פתיחת הקובץ לקריאה מוצב הסמן **בדיסק** על התו הראשון, תו מספר 0. המלה readln מורה למחשב לשלוף מהדיסק את כל התווים מהתו הנוכחי שעליו מוצב הסמן ועד לסוף הרשומה (המסתיימת שם בתווים EOL (end of line) ו-CR (carriage return). הארגומנט במשפט זה מורה למחשב להציב את הרשומה שנשלפה בכתובת **Read1** בגיליון.

2. כתיבת שם הנמען בשורה הראשונה של **Print1**

{put **Print1**,0,0,@mid(**Read1**,61,7)&" "&@mid(**Read1**,68,8)}

הפונקציה @mid שולפת מתוך מחרוזת תווים השוכנת בכתובת **Read1** מספר תווים. הארגומנט הראשון בפונקציה היא הכתובת עם המחרוזת ממנה ישלפו התווים. הארגומנט השני הוא מיקום במחרוזת (0 = התו הראשון משמאל במחרוזת כולה) של התו הראשון **בתת-המחרוזת** לשליפה - תואם למה שקראנו "מספר התו הראשון בשדה" בטופס מבנה הרשומה. והארגומנט השלישי הוא מספר התוים **בתת-המחרוזת** לשליפה - תואם למה שקראנו "גודל השדה".

התפוקה של הפונקציה הזאת בדוגמא שלנו, כשהיא משולבת
ב { ... put} היא ...

1. שליפת השם הפרטי מ-**Read1**.

2. הוספת תו רווח מימין לשם הפרטי.

3. הוספת שם המשפחה מימין לתו הרווח.

4. כתיבת המחרוזת המשולבת הזאת בשורה מספר 0
בעמודה מספר 0 של **Print1**.

3. כתיבת שם הארגון בשורה **השלישית** של **Print1** (כדי לקבל במדבקה שורת רווח בין שם הנמען לשם הארגון).

```
{put Print1,0,2,@mid(Read1,36,25)}
```

4. כתיבת שלוש שורות הכתובת:

```
{put Print1,0,4,@mid(Read1,17,19)}
{put Print1,0,5,@mid(Read1,5,12)}
{put Print1,0,6,@mid(Read1,0,5)}
```

5. הקרנת התפוקות ב-**Print1** (ע"י ~) והפעלת משפט שירות להדפסה

```
~{4}{return}
```

```
4 /rlrPrint1~/pprPrint1~omnouqglq{blank Print1}~{return}
```

כאן מסתיימת הדפסת המדבקה והלוטוס חוזר ומבצע אותן פעולות על רשומת הנמען השני. הקובץ בדיסק עדיין פתוח. הפעולה הקודמת של readln הציבה את הסמן בדיסק על התו הראשון של הרשומה הבאה ולכן יכולה השליפה להמשך ברצף, רשומה אחר רשומה עד ה"מעצור".

כשמונה הפעימות יראה 11, תהיה זו המדבקה האחרונה להדפסה מכיון שהתחלנו מאפס. ב-**Stop1**, כזכור, כתוב 12 (מספר הרשומות בקובץ) ולכן כשהמונה יראה 12 תיפסק ההדפסה והקובץ יסגר.

<u>תרגיל:</u> כתוב Intra-Macro להדפסת מדבקות בדפים בני 8 שורות ושלושה טורים של מדבקות.

<u>רמז:</u> @int((**Counter1**-1)/8)
@mod((**Counter1**-1),8)
ואפשר גם לשלב { ... for} ב-{ ... for}

## 6. כתיבה אל הדיסק - שילוב { ... for} עם Index@

את הקטע הבא בחרתי לצרף בסוף פרק זה, מכיון שיש כאן המשכיות של דרך חשיבה וסגירת מעגל מסוים של שילובי פונקציות, מקרו, עבודה בגיליון ותקשורת ASCII עם הדיסק.

לפני זמן מה כתבתי יישום לוטוס עבור האוניברסיטה העברית בירושלים - המרכז ללימודי חוץ. בזמן הרשמת תלמידים לתכנית "שישי באוניברסיטה" נקלטו הפרטים שלהם באמצעות מסך קלט שהוקם בגיליון לוטוס ונשמרו בקובצי ASCII. אחרי שנקלטו פרטי התלמידים, וגם במהלך הקליטה, צריך להנפיק דוח מודפס המראה את שיבוץ התלמידים לקורסים השונים.

במהלך ביצוע הצבת התלמידים לקורסים השונים נפתח **לכל קורס**, באופן אוטומטי, קובץ ASCII ובו שני נתונים לכל תלמיד ששובץ אליו: מספר ת"ז ומספר רשומה בקובץ התלמידים לאותו קורס.

בעת הכנת הדוח לקורס נשלפת תחילה רשימת התלמידים ששובצו בו. רשימה זו משמשת לשליפת שאר נתוני התלמיד מכתובות שונות בדיסק לשם עיבודם והדפסת הדוח המלא.

מערך היישום כולו נבנה בצורה מודולרית, כמו "לגו", המאפשרת להוסיף בקלות יחסית "משבצות" תוכנה. משבצת אחת כזאת היא מסך לדרישת דיווח על רשימה של קורסים. דהיינו, מקום להקלדת מספרי קורסים שרוצים לקבל שיבוץ עבורם **בלבד** והדפסתם.

הדוגמא הבאה היא חלק ממשבצת התוכנה הזאת הקשורה לנושאים שעסקנו בהם עד עתה.

הקמת היישום מתחילה מהשאלה: "מה רוצה המשתמש לקבל?". ובכן, המשתמש רוצה להקיש מספר של קורס בודד, או מספר קורסים, ללחוץ על כפתור ו"שידפיס לי את השיבוץ לקורסים שביקשתי". נתחיל בהכנה של איזור להקלדת מספרי הקורסים:

A B C D E F G H I J

להקשת מספרי הקורסים
להדפסה

| | | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 1 | 2 | 3 | 4 | 5 |

<—— /rncData2~b7.g16~

בתחום המיועד להקשת מספרי הקורסים נכין שישה טורים של **0**, עשר שורות בכל טור. כלומר, אפשר לבקש הדפסת שיבוץ לקורס אחד ועד רשימה של שישים קורסים.

כדי שתכנית ההדפסה תדע אחר כך איזה קורסים להדפיס, נרשום בקובץ ASCII את מספרי הקורסים שהוקשו אל הלוח שלמעלה. אבל איננו רוצים לרשום אפסים בקובץ הזה. דהיינו, יש צורך לסרוק את כל 60 התאים ב-Data2 ולכתוב אל הקובץ רק אם המספר איננו אפס.

אפשר לבצע מטלה זו במספר דרכים. נדגים כאן אחת מהן:

```
        1 {open FileName2,w}
          {close}
          {open FileName2,a}
          {for Counter2,Start2,Stop2,Step2,2}~{branch 3}
        2 {recalc Print2}~{if Index2="0"}{return}
          {writeln Print2}
          {return}
        3 {close}
          {quit}

  @string(@index(Data2,@int((Counter2-1)/10),@mod((Counter2-1),10)),0) ──┐
                                                                           │
    Index2    <────────────────────────────────────────────────────────────┘
    Print2    <──────────────── @repeat(" ",5-@length(Index2))&Index2
FileName2 +Drive1&Dir1&File2&Ext2
   Drive1 c:\
     Dir1 Ran\WS_5\Data1\
    File2 Tabel_2
     Ext2 .Ran{return}
 Counter2
   Start2   1
    Stop2  60
    Step2   1
```

<u>תרגיל:</u> כתוב Intra-Macro זה בגיליון לוטוס והפעל אותו בְ-step. ראה כיצד מספרי הקורסים משתנים ב-**Print2**.



## פרק 6

# ניהול קבצים ואבטחת מידע

בפרקים הקודמים למדנו שיטות ודרכי חשיבה לשימוש מתקדם בלוטוס. בפרק הנוכחי ניישם את השיטות הללו בנושאים הקשורים בניהול קבצים ובאבטחת מידע. כפי שנעשה בפרקים הקודמים, היישומים שיודגמו כאן פותחו רק עד לשלב הדרוש לצורך הסברת הנושא. הקורא יוכל להכניס שיפורים רבים בדוגמאות הללו תוך הפעלת העקרונות שנלמדו. חלק מהשיפורים יוצעו לקורא בתרגילים במהלך ההסברים.

### 1. לוח משולב - נתונים ונוסחאות

ביישומים רבים נשתמש בגיליון לוטוס המכיל:

1. כותרות לעמודות.
2. כותרות לשורות.
3. נתונים הנקלטים ו/או מעודכנים על ידי הקלדה ישירה.
4. נוסחאות שונות כגון סכומים, ממוצעים, תאריכים והתניות.

את הגיליון הזה שומר בדרך כלל המשתמש בשלמותו ע"י /fs ושולף אותו ע"י /fr.

לעתים קרובות "מעמיסות" הנוסחאות את הגיליון עד למצבים שמניבים את ההודעה memory full. כבר התייחסנו בפרקים הקודמים להבדלים בין מבנה לוח פתוח לבין מבנה לוח סגור.

כל הדוגמאות בפרק זה יהיו במבנה לוח פתוח, כמו בלוח מספר 1:

**לוח מספר 1**

**תקציבי הרשויות לפי סעיף תקציבי - תכנון לעומת ביצוע**

± % : תכנון פחות ביצוע ב-% מתכנון ... - חריגה + עודף

* חריגה מתקציב            תקציבי הרשויות באלפי ש"ח

| הרשות | | סה"כ | | | סעיף א | | | סעיף ב | | |
|---|---|---|---|---|---|---|---|---|---|---|
| שם | סוג | תכנון | ביצוע | ± % | תכנון | ביצוע | ± % | תכנון | ביצוע | ± % |
| * בני גד | ממ | 771 | 873 | -13.2 | 578 | 623 | -7.8 | 193 | 250 | -29.5 |
| גנים | מא | 1645 | 1544 | 6.1 | 1234 | 1123 | 9.0 | 411 | 421 | -2.4 |
| * תמר | ממ | 47 | 55 | -17.0 | 35 | 45 | -28.6 | 12 | 10 | 16.7 |
| * אבן פינה | מא | 7719 | 8962 | -16.1 | 5789 | 6789 | -17.3 | 1930 | 2173 | -12.6 |
| דור | ממ | 1039 | 962 | 7.4 | 779 | 750 | 3.7 | 260 | 212 | 18.5 |

## 1.1 שמירה במבנה wk1

בדוגמא שבלוח מספר 1 מודגשים התווים שניתנים לשינוי על ידי הקלדה, כל השאר מופקים על ידי משוואות. מבט חטוף בלוח מגלה שלמעשה המשוואות בכל השורות בעמודה נתונה זהות לגמרי. אין צורך לשמור את הקובץ עם כל הנוסחאות, מכיון ששמירה כזו מעמיסה את הדיסק בנתונים וערכים מיותרים. היא גם גורמת להאטה משמעותית בזמן שליפת הקובץ ומקטינה את כמות הנתונים שאפשר לנהל בקובץ wk1 אחד.

מספר דרכים להתגבר על בעיה זו:

1. שמור רק את הנוסחה הראשונה בכל עמודה.

2. שמור רק את הנוסחה הראשונה בכל עמודה - מחוץ ללוח.

3. שמור רק את הנוסחה הראשונה בכל עמודה - בתוך משפט מקרו.

עם שליפת הגיליון, יועתקו הנוסחאות על ידי פיסקת מקרו \0.

נדגים את השיטה השנייה. בשיטה זו נוסיף ללוח שלמעלה הגדרות תחומים כפי שנראה בתרשים הבא.

## לוח מספר 1

**תקציבי הרשויות לפי סעיף תקציבי - תכנון לעומת ביצוע**

± % : תכנון פחות ביצוע ב-% מתכנון ... - חריגה + עודף

* חריגה מתקציב     תקציבי הרשויות באלפי ש"ח

| רשות | | סה"כ | | | סעיף א | | | סעיף ב | | |
|---|---|---|---|---|---|---|---|---|---|---|
| שם | סוג | תכנון | ביצוע | ± % | תכנון | ביצוע | ± % | תכנון | ביצוע | ± % |
| בני גד | ממ | | | | 578 | 623 | | 193 | 250 | |
| גנים | מא | | | | 1234 | 1123 | | 411 | 421 | |
| תמר | ממ | | | | 35 | 45 | | 12 | 10 | |
| אבן פינה | מא | | | | 5789 | 6789 | | 1930 | 2173 | |
| דור | ממ | | | | 779 | 750 | | 260 | 212 | |

Data1     Data2     Data3     Data4

התחום הראשון, **Data1**, נמצא בדוגמא זו בעמודה A ומפיק את ההפרש באחוזים בין התכנון לבין הביצוע לכל רשות לגבי **סעיף תקציבי ב.**

נבצע את הפעולות הבאות:

1. נציב את הסמן בתא ריק כלשהו בשורה הראשונה של **איזור תפריטי הפקודות** ונבצע ~~/rncLine1

2. נציב את הסמן בשורה הראשונה של Data1

3. נכתוב את המשוואה במלואה:

   ההבדל באחוזים בין התכנון והביצוע,
   סעיף ב ⟵ (b10-c10)/c10*100

4. נבצע ~/c~Line1

5. נבצע ~/re

התחום השני, **Data2**, נמצא בעמודה D ומפיק את ההפרש באחוזים בין התכנון לבין הביצוע לכל רשות לגבי **סעיף תקציבי א.**



נבצע לשם כך את הפעולות הבאות:

1. נציב את הסמן בתא ריק בשורה השניה של **איזור תפריטי הפקודות** מיד מתחת ל **Line1** ונבצע ~~/rncLine2

2. נציב את הסמן בשורה הראשונה של **Data2**

3. נכתוב את המשוואה במלואה:

   ההבדל באחוזים בין התכנון והביצוע,
   סעיף א ——> (e10-f10)/f10*100

4. נבצע ~c~Line2/

5. נבצע ~re/

התחום השלישי, **Data3**, **משתרע על פני שלוש עמודות: I, H, G**. בעמודה I כתובה משוואה המפיקה את סה"כ התכנון בש"ח לכל הסעיפים התקציביים; בעמודה H - ישנו סה"כ ביצוע לכל הסעיפים ובעמודה G - ההפרש ביניהם באחוזים. כלומר, יש שלוש משוואות בכל שורה ב-**Data3**.

נבצע את הפעולות הבאות:

1. נציב את הסמן בתא ריק בשורה השלישית של **איזור תפריטי הפקודות**, מתחת ל-**Line2**, ונבצע ~~/rncLine3

2. נציב את הסמן בשורה הראשונה של **Data3**, בעמודה G

3. נכתוב את המשוואה במלואה:

   ההפרש באחוזים בין התכנון והביצוע,
   סה"כ ——> (h10-i10)/i10*100

4. נזיז את הסמן עמודה אחת ימינה, אל H ונכתוב את המשוואה עבור סה"כ בש"ח, ביצוע לכל הסעיפים:
   ——> @sum(b10.e10)

5. נזיז את הסמן עמודה אחת ימינה, אל I ונכתוב את המשוואה סה"כ בש"ח, תכנון לכל הסעיפים:
   ——> @sum(c10.f10)

6. נחזיר את הסמן לפינה השמאלית העליונה של **Data3** ונבצע:
   /c{r 2}~Line3~

7. נבצע /re{r 2}~

8. ולבסוף, נבצע /rncLine3~{end}{r}~

התחום הרביעי, **Data4**, מפיק "*" לציון חריגה מתקציב הרשות. המשוואה מחזירה * כשיש חריגה מהתקציב:

```
@if(g10<0,"*","")
```

<——

המשוואה מופעלת ב-Data4.

נבצע נוהל Line4 כפי שביצענו עבור שלושת המשוואות הקודמות.

הסיבה לשמירת משוואות אלו בשורות הראשונות של הגיליון היא הגנה עליהן. כך אפשר למנוע שיבושים במשוואות עקב הזזת משטחים בגיליון ו/או שינויים מבניים אחרים.

עכשיו נכלול בפיסקת המקרו 0\ את המשפט הבא:

```
/wgpd/cLine1~Data1~/cLine2~Data2~/cLine3~Data3~/cLine4~Data4~/wgpe
```

<u>הערה:</u> בלוחות מעורבים, המכילים נתונים ונוסחאות, יש לכלול כנוהל תיקני את wgpe/ כדי להגן על הנוסחאות מפני מחיקה על ידי הקלדה עליהן.

פיסקת המקרו לשמירת הלוח במבנה wk1 תכיל את המשפט הבא:

```
/wgpd{blank Data1}{blank Data2}{blank Data3}{blank Data4}~
```

הנוהל שהוסבר מקטין בצורה משמעותית את גודל הקובץ שנשמר ומאפשר הגדלה משמעותית במספר הרשומות ו/או משוואות שאפשר לכלול בו לפני התרעת memory full.

תפוקת המשוואות נדרשת ממש לפני הדפסת הדוח, או כשרוצים לראות את התוצאה הסופית על המסך. לכן, את משפט המקרו שמייצר את תפוקת המשוואות נפעיל רק כאשר זקוקים לה, ובתוספת נוהל rv/.

```
/wgpd/cLine1~Data1~/rvData1~Data1~/cLine2~Data2~/rvData2~Data2~
/cLine3~Data3~/rvData3~Data3~/cLine4~Data4~/rvData4~Data4~/wgpe
```

הפעלת נוהל rv/ בהפקת תוצרי משוואות מגדילה באופן משמעותי את מספר המשוואות שאפשר להפעיל בגיליון במסגרת מגבלת זיכרון.

למדנו עתה נוהל להפקת משוואות **תחום אחר תחום**. אפשר לבצע את הנוהל הזה בתוך תחום נתון, **מלמעלה למטה**. בעיקרון, נחלק את התחום לשני חלקים, חלקו העליון וכל השאר. נעתיק תחילה את המשוואות אל החלק העליון ונבצע rv/. אחר כך נעתיק אותן אל החלק השני ונבצע rv/. יש דרכים רבות לבצע פעולה זו. נדגים אחת מהן.

נניח שיש לוח ובו רצף שורות של נתונים מוקלדים שהגדרנו על ידי **Data2**. גם נניח שמול כל שורת נתונים כזאת צריכה להיות שורה שנמשכת על פני חמש עמודות, משוואה בכל עמודה. את תחום המשוואות הגדרנו בשם **Data1** וביצענו, כפי שלמדנו קודם, נוהל **Line1**.

פיסקת המקרו שתניב את התפוקה המבוקשת תהיה:

```
{goto}Data1~/cLine1~.{d @rows(Data2)/2}~/rvData1~Data1~{end}{d 2}
/cLine1~.{end}{r}{31}{l 4}~/rvData1~Data1~
```

כאשר **31** הוא משפט השירות האוניברסלי:

```
{end}{d 2}{end}{u}{return}
```

גם אם ננהג בנוהל זה, ייתכן שגם הוא לא יספיק ו"הזיכרון ייגמר" לפני שכל הפעולות התבצעו. במקרים רבים ניסוי קצר פותר את הבעיה: בודקים בכמה שלבים צריך לבצע rv/ כדי שאפשר יהיה להפעיל את המשוואות כולן במסגרת הזיכרון הנתון. אחר כך מבצעים זאת במספר שלבים. בדרך כלל (אך לא תמיד) מספיק לבצע זאת בשלושה שלבים: על החצי הראשון של השורות, על החצי הראשון מתוך החצי שנותר ועל כל מה שנשאר.

אולם, גם נוהל זה מוגבל, בסופו של דבר, בגודל הגיליון ובגודל הזיכרון. כדי לבצע עיבודים על כמות אין סופית של נתונים, יש להקים את היישום על פי הנוהלים והעקרונות שבפרק 5.

## 1.2 שמירה במבנה ASCII

לשמירת קבצים במבנה ASCII מספר יתרונות.

במובן מסוים אפשר להתייחס אל גיליון לוטוס כאל קובץ המחזיק תוכנה. לצורך המשך הדיון נגדיר **תוכנה:**

"כל מה שיש בגיליון wk1 **פרט לנתונים**".

תוכנה במבנה של Intra-Macro היא מערך פיסקאות מקרו קומפקטי ויעיל והקובץ שמחזיק אותה יהיה בדרך כלל קטן מאוד (אפשר לבצע מספר רב ביותר של מטלות ב Intra-Macro שמחזיק 10-15 kb). ביישומים רבים אפשר לחלק מטלות בין מספר קובצי Intra-Macro ולהפעילם אוטומטית. דהיינו, כשקובץ 1.wk1 מסיים את ביצועיו מתבצע ~fr2.wk1/ הקובץ שנשלף מכיל מקרו 0\ ולפיכך מתחיל לפעול מיד עם טעינתו. כך נוצרת **שרשרת** של קובצי לוטוס. בשרשרת כזאת אפשר לשלב מערכת קובצי תוכן ענינים (ראה פרק מספר 1) לאינטראקציה בין האדם והמחשב.

כאשר במערך כזה נשמרים הנתונים במבנה ASCII היישום כולו מהיר ויעיל. לכל Intra-Macro יש גישה אל כל הנתונים לשליפה, לעיבוד ולשמירה וקובצי התוכנה קטנים ומהירים. חלוקת המטלות להפעלה בשרשרת

מאפשרת ביצוע שינויים בתוכנה בלי לשבש חלקי תוכנה אחרים. בתוך הקובץ עצמו נותן השילוב במקרו של פקודות התפריט ופונקציות המחרוזת הפעלה מהירה במבחר מגוון של מטלות.

בפרק זה נתייחס לניהול התוך גיליוני בלבד של שילוב ASCII בגיליון - ובעיקר בנוהל pf/ ו-fi-/.

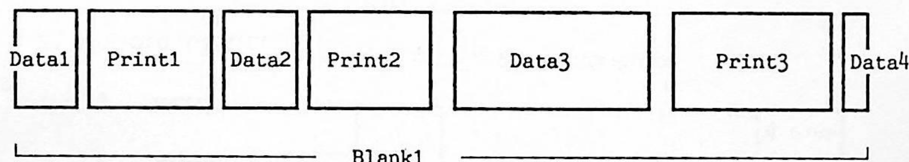
נוהלי השמירה והשליפה מקובצי ASCII פשוטים למדי. כל הדרוש הוא הגדרת שלושה משטחים (מטריצות) נוספים בלוח מספר 1, שיראה עכשיו (בטרם שליפה) כך:

**לוח מספר 1**

**תקציבי הרשויות לפי סעיף תקציבי - תכנון לעומת ביצוע**

± % : תכנון פחות ביצוע ב-% מתכנון ... - חריגה + עודף

| * חריגה מתקציב | | תקציבי הרשויות באלפי ש"ח | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| הרשות | | סה"כ | | | סעיף א | | | סעיף ב | | |
| שם | סוג | תכנון | ביצוע | ± % | תכנון | ביצוע | ± % | תכנון | ביצוע | ± % |




לשמירת הנתונים נכתוב את פיסקת המקרו הבאה:

```
/wgpd{open FileName1,w}
{close}
/pfX{esc}{FileName1}~rrPrint1~omnouqgq
{open FileName2,w}
{close}
/pfX{esc}{FileName2}~rrPrint2~omnouqgq
{open FileName3,w}
{close}
/pfX{esc}{FileName3}~rrPrint3~omnouqgq{blank Blank1}~
{quit}
```

בפיסקה זו בולטת העובדה **שהנתונים לא נשמרים בקובץ אחד**. נוהל זה מכין אותנו לקראת הדיון **במידור פונקציונלי** בהמשכו של פרק זה.

כדי לשלוף את הנתונים נכתוב את פיסקת המקרו:

```
/wgpd
{goto}Print1~/finX{esc}{FileName1}~
{goto}Print2~/finX{esc}{FileName2}~
{goto}Print3~/fitX{esc}{FileName3}~
/wgpd/cLine1~[1mData1~/cLine2~Data2~/cLine3~Data3~/cLine4~Data4~
/wgpe{quit}
```

וכפי שלמדנו בפרק 3, נפעיל את הגיליון ע"י תפריט פקודות.

## 2. מידור פונקציונלי

אפשר לאמר שכל משתמש לוטוס הוא, בו-זמנית, מערכת אוטונומית שמבצעת מטלות שונות ותת-מערכת של מערכת-על. נתייחס כאן אל המשתמש כמערכת אוטונומית. ובדוגמא שלנו הוא נדרש לבצע פעולות שונות על מכלולים מתוך מערך נתונים עם מבנה פשוט יחסית:

1. סעיף תקציבי מספר N [1,2,...,N]

2. סטטוס תקציבי

   1. תכנון

   2. ביצוע

3. רשויות מקומיות

   לפי רשימה נתונה

את כל המידע הזה אפשר לארגן במדורים בהתאמה לצרכי המשתמש. שמירת נתונים בקובץ wk1 נותנת מידור מינימלי, כי כאשר צריך לפעול על נתון כל שהוא, או על מכלול מתוך כלל הנתונים, הכל נמצא בכתובת אחת והנגישות הינה דרך קובץ wk1 אחד בלבד. שמירת הנתונים במבנה ASCII היא תחילתו של תהליך המידור הפונקציונלי.

במידור נתונים פונקציונלי תתבצע שמירת נתונים במבנה מודולרי. מטריצות נתונים יוגדרו על פי צורכי המשתמש בזמן הנוכחי ולפי תחזית הצרכים בעתיד אם יחידת המידע הקטנה ביותר היא עמודת נתונים אחת וכל שורה בעמודה זו מתייחסת לרשות מקומית אחרת, אזי מידור מינימלי מקבלים כאשר כל הנתונים בלוח נשמרים בכתובת אחת. מידור מקסימלי יהיה כאשר נשמור כל עמודה בכתובת נפרדת (קובץ ASCII). מידור אופטימלי מקבלים כאשר מחלקים את המידע בכתובות לפי צורכי השימוש בו. לשם הדגמה נניח מערך מטריצות נתונים שמור בקובצי ASCII.

### מבנה הלוח בגיליון לוטוס

| סעיף N | | | | סעיף ב | | | | סעיף א | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| % | ± | ביצוע | תכנון | % | ± | ביצוע | תכנון | % | ± | ביצוע | תכנון |

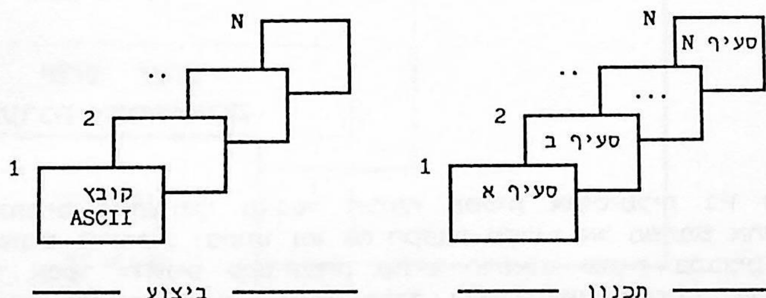
| | | | | | |
|---|---|---|---|---|---|
| Ptint_2_N | Ptint_1_N | Ptint_2_2 | Ptint_1_2 | Ptint_2_1 | Ptint_1_1 |

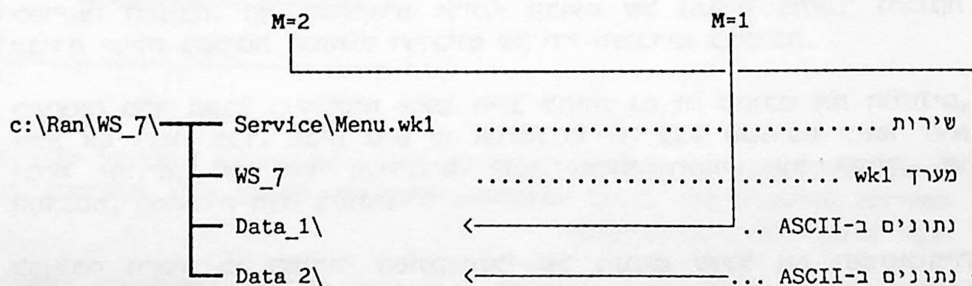
הלוח מאורגן כך שאפשר להוסיף בו סעיפים תקציביים כפי הצורך. כל עמודת נתונים מוגדרת בשם תחום משלה, כאשר שם התחום מורכב מהמלה **Print** ומשני אינדקסים. האינדקס הראשון משמאל מסמל את הסטטוס התקציבי (1=תכנון, 2=ביצוע) והאינדקס השני מסמל את הסעיף התקציבי (החל ב-1 ועד N).

הנתונים הללו מאוחסנים בקובצי ASCII בדיסק במבנה תואם את המבנה בגיליון.

### מיפוי ל-Meta-Macro

שם הקובץ: c:\Ran\WS_7\Data_M\N.Ran



כתובת ב-Intra-Macro

```
   Drive1 c:\
     Dir1 Ran\WS_7\
  SubDir1 Data_
    File1 +"\"&NN1
      NN1       <—— left aligned label     מספר הסעיף התקציבי
      MM1       <—— left aligned label  (1 או 2) סטטוס הנתונים
     Ext1 .Ran{return}
FileName1 +Drive1&Dir1&SubDir1&MM1&File1&Ext1
```

כאשר NN1=2 ו- MM1=1 עבור "נתוני תכנון בסעיף ב'" נקבל ב-**FileName1**:

```
c:\Ran\WS_7\Data_1\2.Ran{return}
```

<u>תרגיל:</u> כתוב פיסקת מקרו לשמירת הנתונים של N סעיפים תקציביים המופעל על ידי {for ... } ואשר בו **NN1** ו-**MM1** הם פונקציות של **Counter1** (מונה הפעימות של {for ... }).

כתוב פיסקת מקרו כנ"ל לשליפת הנתונים ושלב בה נוהל **Line1**.

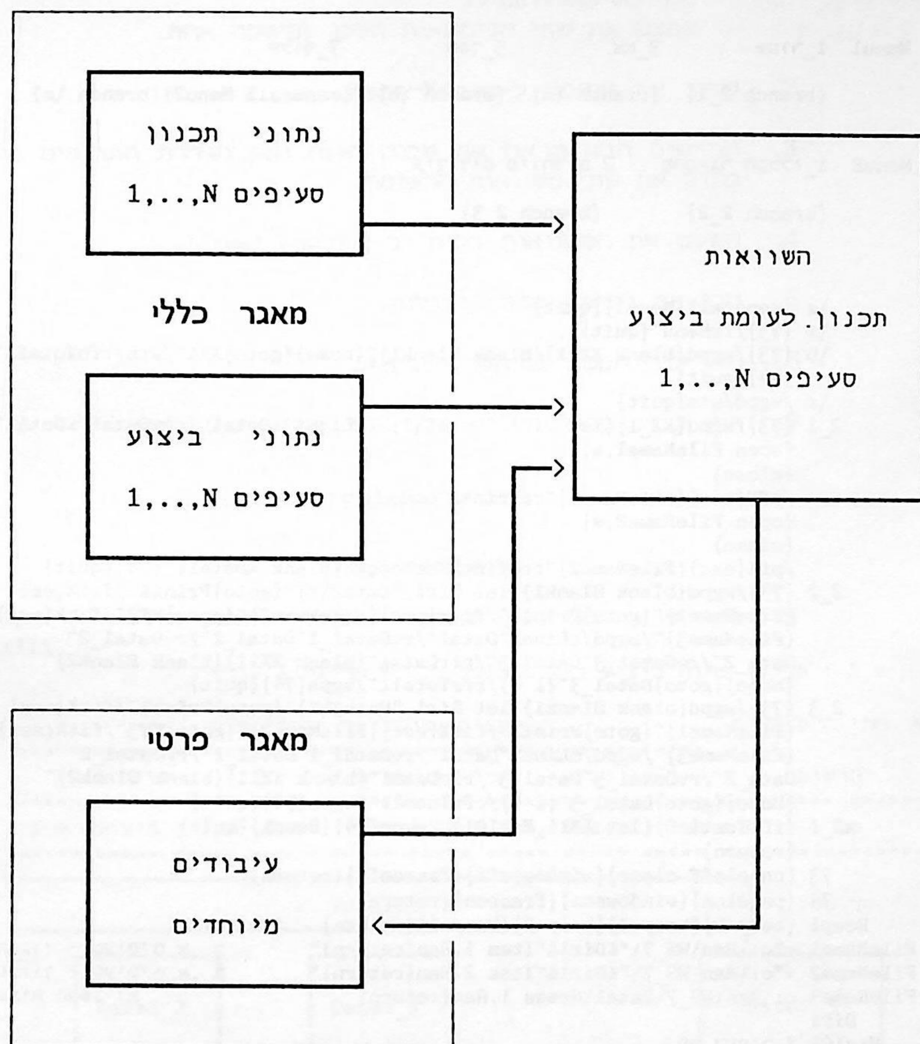
## 3. מערכת אינטגרטיבית

מידור נתונים פונקציונלי מאפשר חלוקת מטלות אופטימלית בין משתמשי לוטוס שונים במערכת, ובאותו זמן גם הקבצת מטלות אל משתמש אחד. ותוך כדי כך אפשר להעניק למשתמשים שונים הרשאות שונות בקבצים שונים. למשל, ניתן להעניק הרשאה לקריאה בלבד, הרשאה לעדכון בלבד, או הרשאה לקריאה וכתיבה. כך מתאפשרת זרימה קבועה של נתונים במאגר וחלוקת עבודה יעילה בקליטת הנתונים ועדכונם על ידי העובדים במערכת.

כתוצאה מכך מקבל ההחלטות איננו חייב להיות גם זה שקולט את הנתונים, אלא אם רצונו בכך. מכיון שיש לו זכויות קריאה **בכל המערכת** כאשר הוא יבקש נתונים, הם יהיו מעודכנים בזמן אמיתי ויופיעו על המסך, או המדפסת, במהירות רבה ביותר.

לכאורה נראה, כי המידור הפונקציונלי של נתונים שולל את האוטונומיה של המשתמש ואת אי תלותו באחרים, אך לא כך הוא הדבר. התרשים הבא מדגים כיצד, בסביבה הנשלטת על ידי לוטוס, אפשר ליהנות מיתרונות המערכת יחד עם שמירה על אוטונומיה מירבית. או, במלים אחרות, מה שנכנה בהמשך - **מערכת אינטגרטיבית.**

## מערכת אינטגרטיבית




ה-Intra-Macro הבא מתייחס למבנה של לוח 1 ומדגים את הטכניקה לשימוש במערכת אינטגרטיבית באופן אוטומטי. פיסקאות המקרו כולן מבוססות על חומר שנלמד בפרקים הקודמים.

תפריט פקודות המסתעף אל תפריט נוסף נותן למפעיל בחירה בין המאגר הפרטי לבין המאגר הכללי, ובאותה עת, שליטה אוטומטית בשליפת ושמירת נתונים.

```
Menu1  1_שמור          3_צא          5_מסך          7_שלוף

       {branch 2_1}  {branch \m}  {branch \b}  {menucall Menu2}{branch \z}

Menu2  1_מהמאגר הכללי      2_עיבודים מיוחדים

       {branch 2_2}        {branch 2_3}

        \z {menucall Menu1}{quit}
        \m {73}/frMenu~{quit}
        \b {73}/wgpd{blank XX11}{blank Blank1}~{home}{goto}XX1~/wtb/rfhTotal1~
           {74}{quit}
        \a /wgpd/wtc{quit}
       2_1 {73}/wgpd{x2_1}{let Dir1,"Data3\"}~/cLine1~xData1~/rvxData1~xData1~
           {open FileName1,w}
           {close}
           /pfX{esc}{FileName1}~rrPrint1~omnouqgq
           {open FileName2,w}
           {close}
           /pfX{esc}{FileName2}~rrPrint2~omnouqgq{blank xData1}~{74}{quit}
       2_2 {73}/wgpd{blank Blank1}{let Dir1,"Data2\"}~{goto}Print1~/fitX{esc}
           {FileName1}~{goto}Print2~/fitX{esc}{FileName2}~{goto}YY3~/fitX{esc}
           {FileName3}~/wgpd/cLine2~Data1~/rvData1_1~Data1_1~/rvData1_2~
           Data_2~/rvData1_3~Data1_3~/rlrData1~{blank XX11}{blank Blank2}~
           {home}{goto}Data1_3~{l 4}/rfrTotal1~/wgpe{74}{quit}
       2_3 {73}/wgpd{blank Blank1}{let Dir1,"Data3\"}~{goto}Print1~/fitX{esc}
           {FileName1}~{goto}Print2~/fitX{esc}{FileName2}~{goto}YY3~/fitX{esc}
           {FileName3}~/wgpd/cLine2~Data1~/rvData1_1~Data1_1~/rvData1_2~
           Data_2~/rvData1_3~Data1_3~/rlrData1~{blank XX11}{blank Blank2}~
           {home}{goto}Data1_3~{l 4}/rfrTotal1~/wgpe{74}{quit}
      x2_1 {if Test1=0}{let XX11,Msg101}~/wgpe{74}{Beep1}{quit}
           {return}
        73 {paneloff clear}{windowsoff}{frameoff}{return}
        74 {panelon}{windowson}{frameon}{return}
     Beep1 {beep 2}{beep 1}{beep 2}{beep 1}{return}
 FileName1 +"c:\Ran\WS_7\"&Dir1&"Item 1.Ran{return}"        תכנון - סעיפים א, ב
 FileName2 +"c:\Ran\WS_7\"&Dir1&"Item 2.Ran{return}"        תכנון - סעיפים א, ב
 FileName3 c:\Ran\WS_7\Data1\Names 1.Ran{return}             שמות הרשויות
      Dir1
    Msg101 ? מה לשמור
```

בהמשך יוסבר המיפוי של Intra-Macro זה. אבל תחילה מומלץ לפתור את התרגילים הבאים.

<u>תרגיל:</u> 1. בין פיסקה **2_2** לבין פיסקה **2_3** יש הבדל **אחד בלבד**.
- מהו ההבדל?
- מהי הסיבה להבדל?
- צמצם את שתי הפיסקאות הללו לפיסקה אחת.

2. שחזר את המיפוי ל-Meta-Macro.

3. התרשים הבא מראה את מבנה הלוח ואת הגדרת התחומים בו. כתוב את שתי משוואות הרשומה.

4. השלם את המשוואות בלוח וב-**Line1** ו-**Line2**.

5. הכן את רשימת סדר הפעולות.

6. הכן לוח הקמת Menu1 ו-Menu2.

7. הכן טופס מבנה רשומה לכל אחד משלושת הקבצים.

**לוח מספר 1**

**תקציבי הרשויות לפי סעיף תקציבי - תכנון לעומת ביצוע**

XX11

± % : תכנון פחות ביצוע ב-% מתכנון ... - חריגה + עודף

* חריגה מתקציב תקציבי הרשויות באלפי ש"ח

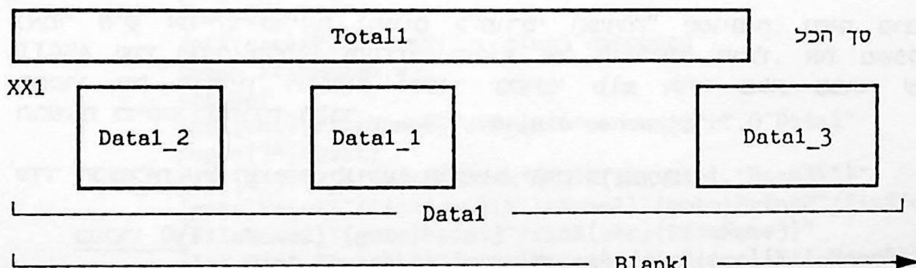
==================================================

הרשות סה"כ סעיף א סעיף ב

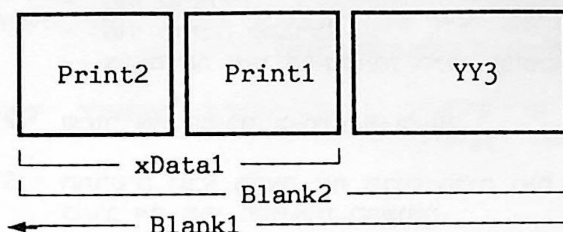
================ ================ ================ ================

שם סוג תכנון ביצוע ± % תכנון ביצוע ± % תכנון ביצוע ± %

===== ===== ===== ===== ===== ===== ===== ===== ===== === ============



מימין ללוח מוגדרים התחומים הבאים:




## 4. מידור מאבטח

לנתונים, כמו למשאבים אחרים של הארגון, יש ערך. כשניגשים לאבטח דבר ערך מקימים מערך אבטחה בהתאם למחיר של אובדן אותו דבר ערך. אפשר להחביא נתונים "מתחת למזרון", אפשר לשמרם מאחרי מנעול ובריח (מלת מפתח - password) ואפשר להציב שומרים ומערכות אזעקה סביבם.

דרך נוספת לאבטחת נתונים היא על ידי מידור. למעשה, המידור הפונקציונלי המירבי הוא שלב ראשון באבטחת מידע. את הקבצים אפשר לפזר במקומות שונים בדיסק ועל דיסקטים. מכיון שכל עמודה נמצאת בקובץ נפרד **ללא כותרות** אין משמעות לנתונים, אלא עד שמקבצים אותם אל הגיליון המפעיל.

כך הופך הגיליון המפעיל למפתח לפענוח מאגר המידע.

בעבודה עם מלת המפתח יכולה לחול תקלה, כי אפשר לשכוח אותה. נחפש על כן דרכים נוספות ופשוטות לאבטחה מפני הפעלה לא חוקית של הגיליון.

נאמר שיש זכויות קריאת נתונים ל"מורשי הפעלה" מסוימים. נקים קובץ ASCII שבו יהיו מספרי תעודות הזהות של המורשים הללו. את משפטי המקרו של הגיליון המפעיל נשמור בקובץ mlb תחת מלת מפתח עם הרשאת כניסה למתכנת בלבד.

סדר הפעולות של הכניסה לגיליון המפעיל יהיה כדלקמן:

<u>במקרו 0\</u>:

1. כתוב לקובץ ASCII את התאריך והשעה של הפעלת הגיליון.
2. דרוש מספר תעודת זהות.
3. בדוק שהמספר קיים ברשימת מורשי הפעלה.
4. רשום את המספר ליד התאריך והשעה של הפעלת הגיליון.
5. אם מורשה הפעלה - תן זכויות לפי המוגדר לאותו מורשה.
6. אם איננו מורשה - בצע פעולות אזעקה.
7. כתוב תאריך ושעת יציאה מהקובץ.

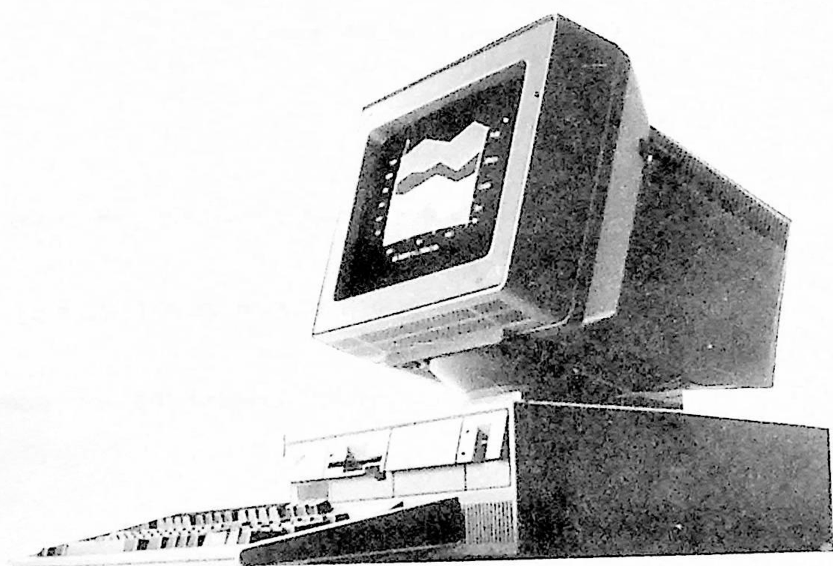
המקרו הבא מציג נוהל לאבטחת מידע על ידי רשימת מורשי הפעלה. המקרו ייכתב ללא כל הסברים, מכיון שכל מרכיביו הוסברו כבר והקורא יכול, כתרגיל, להקים Meta-Macro ו-Intra-Macro ללא קושי.

ניתן לבצע במקרו זה שיפורים רבים. הקורא המעוניין יפיק תועלת מניתוח אפשרויות השיפור ומביצוען.

```
Menu1   שמור_1            צא_3            מסך_5            שלוף_7
        {branch 2_1}  {branch \m}  {branch \b}  {branch 2_2}

    \z {menucall Menu1}{quit}
    \m {73}/frMenu~{quit}
    \b {73}/wgpd{blank XX11}{blank Data1}~/rf,0~Data1~{home}
       {goto}Data1~/wtb/rfhTotal1~{74}{quit}
    \a /wgpd/wtc{quit}
   2_1 {73}/wgpd{x2_1}{2_3}/rfrData1~{let Dir1,"Data3\"}~
       {open FileName1,w}
       {close}
       /pfX{esc}{FileName1}~rrPrint1~omnouqgq
       {open FileName2,w}
       {close}
       /pfX{esc}{FileName2}~rrPrint2~omnouqgq
       {open FileName3,w}
       {close}
       /pfX{esc}{FileName3}~rrPrint3~omnouqgq
       {let Dir1,"Data4\"}~{open FileName4,w}
       {close}
       /pfX{esc}{FileName4}~rrPrint4~omnouqgq
       {open FileName5,w}
       {close}
       /pfX{esc}{FileName5}~rrPrint5~omnouqgq
       {open FileName6,w}
       {close}
       /pfX{esc}{FileName6}~rrPrint6~omnouqgq/rf,0~Data1~
       /wgpe{74}{quit}
   2_2 {73}{2_3}{2_4}/wgpd{blank XX11}{let Dir1,"Data3\"}~
       {goto}Print1~/finX{esc}{FileName1}~{goto}Print2~/finX{esc}
       {FileName2}~{goto}Print3~/finX{esc}{FileName3}~
       {let Dir1,"Data4\"}~{goto}Print4~/finX{esc}{FileName4}~
       {goto}Print5~/finX{esc}{FileName5}~{home}{goto}Print5~
       /rfrTotal1~/rf,0~Data1~/rf,0~Total1~{goto}YY7~/fitX{esc}
       {FileName7}~/cLine6~Data6~/rvData6~Data6~/rlrData6~
       {blank Blank7}~/wgpe{home}{goto}Print1~{74}{quit}
```

```
       2_3 {open FileName8,m}{open FileName8,w}
           {setpos 0}
           {readln Read8}
           {setpos 0}
           {write @string(@now,4)}
           {close}
           {let Read8,@value(Read8)}~
           {return}
       2_4 /wtc/wgpd{getlabel Msg51,xZZ9}~{goto}Blank9~/fitX{esc}
           {FileName9}~/dqiData9~cZZ9~f
           {if @cellpointer("contents")=xZZ9}{esc}q{blank Blank9}{home}{goto}Data1~/wtb/wgpe{74}{return}
           q{open FileName10,a}{open FileName10,w}
           {writeln +"Id= "&xZZ9&" Time= "&@string(@now,4)}
           {close}
           {blank Blank9}{home}{goto}Data1~/wtb/wgpe
           {let XX11,Msg102&xZZ9}~{74}{Beep2}{wait @now+@time(0,0,5)}
           {branch \b}
      x2_1 {if Test1=0}{let XX11,Msg101}~/wgpe{74}{Beep1}{quit}
           {return}
        73 {paneloff clear}{windowsoff}{frameoff}{return}
        74 {panelon}{windowson}{frameon}{return}
       ZZ9 x1
      xZZ9
     Read8
     Beep1 {beep 2}{beep 1}{beep 2}{beep 1}{return}
     Beep2 @repeat(@left(Beep1,16),12)&"{return}"
 FileName1 +"c:\Ran\WS_7\"&Dir1&"1_1.Ran{return}"            סעיף א
 FileName1 +"c:\Ran\WS_7\"&Dir1&"1_2.Ran{return}"            סעיף ב
 FileName3 +"c:\Ran\WS_7\"&Dir1&"1_3.Ran{return}"            סעיף ג
 FileName4 +"c:\Ran\WS_7\"&Dir1&"1_4.Ran{return}"            סעיף ד
 FileName5 +"c:\Ran\WS_7\"&Dir1&"1_5.Ran{return}"            סעיף ה
 FileName6 c:\Ran\WS_7\Data4\1_6.Ran{return}              סה"כ
 FileName7 c:\Ran\WS_7\Data1\Names_1.Ran{return}          שמות הרשויות
 FileName8 c:\Ran\WS_7\Data1\X_1.Ran{return}          דיווח כניסה אחרונה
 FileName9 c:\Ran\WS_7\Data1\X_2.Ran{return}          רשימת מורשי הפעלה
FileName10 c:\Ran\WS_7\Data1\X_3.Ran{return}        דיווח נסיון כניסת נפל
      Dir1
     Msg51             <- מספר תעודת הזהות שלך
    Msg101 ? מה לשמור
    Msg102 ...לא מורשה הפעלה, נשלחה הודעה למרכז הבקרה
```



# פנייה למשתמשי לוטוס

המחבר מציע לקוראים להשתתף בסדנאות המומחים בלוטוס, ולהשיב ע׳ השאלון המצורף.

## סדנת חשיבה אינטגרטיבית בהפעלת גיליון לוטוס

**Intra-Macro ו- Meta-Macro**

מנחה: **רן אברהמי MBA LAC**

ספר לימוד: **מקרו למתקדמים בלוטוס - Meta-Macro ו-Intra-Macrto**
**מאת: רן אברהמי, הוצאת הוד-עמי.**

משך הקורס: **24 שעות לימוד בארבע פגישות שבועיות בנות 6 שעות כל אחת.**

ערכה לתלמיד: **בסיום הקורס יקבל כל תלמיד דיסקטים עם כל המקרו שנלמדו.**

### הנושאים הנלמדים בקורס

**1. ניהול סביבות עבודה ושילוב wk1 <—> ASCII**

הקמת תוכן עניינים במקום fr/ והסתעפויות; הפעלת תוכנות שכנות; הלוטוס כ-Editor; מיפוי הדיסק לפני הפעלת יישומי לוטוס; אוטומציה של הכניסה ללוטוס; מודולריות וארגומנטים ב-Intra-Macro; פקודות pf/ ו-fi/ לשליטה בקובץ ASCII; Data Parse; פונקציות String.

**2. שליטה אוטומטית בפעילות "תוך גיליונית"**

גיליון במבנה Intra-Macro; שליטה אוטומטית ע"י מקרו {menu...} בשילוב ri/ תחת מקרו; עץ תפריטים; נתונים ב-ASCII; משפט מפעיל ומשפט שירות, מודולריות וארגומנטים מורכבים, פונקציות במקרו, טקסט במקרו ע"י Msg; ספריית מקרו וטעינה אוטומטית.

### 3. שליטה אוטומטית בתווים בודדים וברשומות בודדות בדיסק

שליטה בתו, ברשומה, ברשומות בדיסק; הפעלה בדיסק ע"י המקרו {open...} בשילוב המקרו {for...}; הפקת רשומת ASCII במבנה Data Base מנתונים מעורבים (טקסט ומספרים); שילוב {put...} עם @index; עבודה עם קובץ ASCII אין סופי מבלי לייבא אותו אל הגיליון; Data Base רב ממדי ב-ASCII על הדיסק ושליטה אוטומטית מהגיליון:

### 4. ניהול קבצים, אבטחת מידע ונושאים נבחרים

פתרונות לתפעול קבצים גדולים מאוד: הגנה על נוסחאות בעת הקשת נתונים; /ri עם /wt להקלדה מבוקרת; @if(x,"","*") לסימון נתון קריטי; @repeat לבניית גרף דינמי; @vlookup לשליפת ערכים מהדיסק; Data Query; רגרסיות; לוח גאנט; מידור פונקציונלי; מידור מאבטח; דיווח אוטומטי למרכז בקרה; רשימת מורשי הפעלה; הגנה מהעתקת יישום לוטוס.

<u>**שאלון למשתמשי לוטוס ותיקים**</u>

לכבוד רן אברהמי LAC
ת.ד. 4305 ירושלים 91042
טלפון: 02-631768

אני מעונין במידע נוסף בנושאים הבאים:

1. נושאים הנלמדים **בקורס לחשיבה אינטגרטיבית:**

   נושא מספר 1 ____ נושא מספר 2 ____ נושא מספר 3 ____

   נושא מספר 4 ____ כל הקורס ____

2. **שולחן עגול:** מפגש עם עמיתים בהנחייתך לדיון פתוח בנושאים שוטפים שמעסיקים את המשתתפים ____

3. **מטלוג:** פרסום תקופתי לדו-שיח בין משתמשי גיליון מתוחכמים ברמה גבוהה ____

4. **קו-חם למקרו:** שאלות בטלפון בנושאים **מתקדמים** בהפעלת הגיליון האלקטרוני **ובכתיבת מקרו** ____

5. **נושאים אחרים**, אנא ציין: ____________________

____________________

**—— פרטים אישיים ——**

שם: משפחה ____________ פרטי ____________

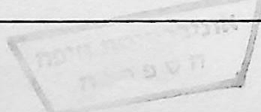
מקום עבודה ____________________

תפקיד: ____________ אגף/מחלקה ____________

כתובת בעבודה: רחוב ומספר ____________________
ישוב ____________ מיקוד ______
טלפון ____________ פקס ____________

בבית: רחוב ומספר ____________________
ישוב ____________ מיקוד ______
טלפון ____________ פקס ____________

איזה תוכנות מכיר? ____________________
איזה גירסאות לוטוס מכיר? ____________________
כמה זמן כבר משתמש בלוטוס? ____________________
באיזו גירסה משתמש עכשיו? ____________________



# רשימות